Information Sciences and Technologies Bulletin of the ACM Slovakia

Association for Computing Machinery

Advancing Computing as a Science & Profession

December 2020
Volume 12, Number 2

Aim and Scope of the Information Sciences and Technologies Bulletin of the ACM Slovakia

ACM Slovakia offers a forum for rapid dissemination of research results in the area of computing/informatics and more broadly of information and communication sciences and technologies. It is primarily a web based bulletin publishing results of dissertations submitted at any university in Slovakia or elsewhere, perhaps also results of outstanding master theses. Besides that, conferences that meet bulletin's expectations with regard to scientific rigor are invited to consider publishing their papers in the bulletin in form of special issues. Besides the web version of the bulletin, a paper version is available, too.

The Bulletin aims:

- To advance and to increase knowledge and interest in the science, design, development, construction, languages, management and applications of modern computing a.k.a. informatics, and more broadly of information and communication sciences and technologies.
- To facilitate a communication between persons having an interest in information and communication sciences and technologies by providing a forum for rapid dissemination of scholarly articles.

Scope of the Bulletin is:

- original research in an area within the broader family of information sciences and technologies, with a particular focus on computer science, computer engineering, software engineering and information systems, and also other similarly well established fields such as artificial intelligence or information science.

Types of contributions:

- **Extended abstracts of doctoral dissertations.** This is the primary type of article in the Bulletin. It presents main contributions of the dissertation in form of a journal paper together with separate section with list of published works of the author. In Slovakia and the Czech Republic, it corresponds to typical length of so called *autoreferat*. In fact, it is envisaged that publishing the extended abstract in the Bulletin makes *autoreferat* obsolete and eventually can replace it completely. It should be noted that by publishing it in the Bulletin, the extended abstract will receive a much wider dissemination. Exceptionally, at the discretion of the Editorial Board, the Bulletin may accept extended abstracts of other than doctoral theses, e.g. Master theses, when research results reported are sufficiently worthy of publishing in this forum. Rules and procedures of publishing are similar.
- **Conference papers.** The Bulletin offers organizers of interesting scientific events in some area within the scope of the Bulletin to consider publishing papers of the Conference in the Bulletin as its special issue. Any such proposal will be subject of discussion with the Editorial Board which will ultimately decide. From the scientific merit point of view, method of peer reviewing, acceptance ratio etc. are issues that will be raised in the discussion.

Besides that the Bulletin may include other types of contributions that will contribute to fulfilling its aims, so that it best serves the professional community in the area of information and communication sciences and technologies. There are four regular issues annually.

Editorial Board

Executive Editor: *Dominik Macko*
Cover Design: *Peter Lacko*
Typeset in LaTeX using style based on ACM SIG Proceedings Template.

Improving Resource Management in Software Defined Networks

Marek Galinski*
Institute of Computer Engineering and Applied Informatics
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
marek.galinski@stuba.sk

Abstract

For real-time multimedia sessions, it is not possible to perform per-packet load balancing as these services are sensitive to delay and the delivery order of individual fragments. These data flows must be considered unsplittable and therefore we must use the load-balancing mechanism on a per-flow basis. If the sum of the free transmission capacities of all paths is greater than the transmission capacity required by this new data flow, there is a possibility that by rearranging the existing data flows between paths, sufficient free capacity will be available on one of the paths to meet the requirements of the new data flow while avoiding congestion. In this work, we express this problem by using the Knapsack problem, respectively by defining a new type of problem by combining existing known subtypes. This view allowed us to design the QFLA algorithm, looking for an improved solution reallocations are needed. However, unlike other known solutions, the main criterion in our case is not maximizing capacity but minimizing calculation time and minimizing the number of reallocations required to achieve an improved solution. The proposed algorithm was verified in two ways - by simulating the calculations themselves with different numbers and sizes of backpacks and objects, but also by simulating the implementation of the algorithm in the SDN environment.

Categories and Subject Descriptors

C.2.0 [**Networks**]: General; C.2.1 [**Networks**]: Network Architecture and Design; C.2.3 [**Networks**]: Network Operations

*Recommended by thesis supervisor: Prof. Ivan Kotuliak Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on June 26, 2020.

Galinski, M. Improving Resource Management in Software Defined Networks. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 12, No. 2 (2020) 1-7

Keywords

Software Defined Networking, Load Balancing, Knapsack Problem, Optimization Problem, Network Monitoring, Network Management

1. Introduction

Multimedia over Internet protocol (usually covered also as Voice over IP) has developed as a mainstream platform. SIP protocol has been standardized by IETF in RFC3261 [20] and is commonly accepted as a standard. Even the 3GPP has adopted it in IMS (IP Multimedia Subsystem) as a signalling protocol (with some further information in RFC4083 [8]). Despite this technology is mature and widely used it still faces problems, among which are load balancing and quality of service of multimedia sessions in multipath VoIP networks. IP is designed as best-effort protocol, it does not implement or guarantee any kind of Quality of Service mechanism.

Standard approaches of VoIP architectures in similar scenarios are not optimal, when it comes to load balancing and quality management and this problem has been analyzed and solved by many reserearch teams throughout the years (for example in [23]). As a solution we propose a new architecture capable of active management of ongoing multimedia sessions.

Software Defined Networks (SDN) and Network Function Virtualization (NFV) in the world of computer networks open new possibilities for managing and optimizing network traffic. The load balancing mechanisms are well known since years ago, and they are naturally part of the SDN environment. Nevertheless, most of these mechanisms are insufficient these days, since most of the current network traffic consists of real time multimedia sessions. These data flows cannot be (or at least are not supposed to be) load balanced per-packet. In the same manner, it is not sufficient to handle congestion after it happens, instead of avoiding it to happen. This thesis sets its objective to continue to work on the Quick Forward Lookup Algorithm (QFLA) brought in [7] in the way of extending the algorithm for the SDN environment, since it was intended for use in the SIP Single Port networks, and to evaluate and formalize the QFLA, comparing performance of the QFLA with existing load balancing mechanisms in SDN environment.

2. State of the Art

The Internet itself is designed in the manner, where the end nodes are responsible for most of the tasks concern-

ing the communication. The network by design does not actively maintains the quality of service, nor handles the congestion or other impairments. This design has two main advantages: Firstly, it allows a unified best-effort service for any type of data at the network layer where service definitions are made at the upper layers (hosts). Secondly, it reduces the overhead and the cost at the network layer without losing reliability and robustness. [11]

This type of architecture fits perfectly to data transmission where the primary requirement is the delivery of the data. Yet, in multimedia transmission, timely delivery is crucial. Multimedia streaming applications have strict delay requirements which cannot be guaranteed in the best-effort Internet. [11]

2.1 Differentiated Services

Differentiated services (DiffServ) is an architecture, that specifies simple mechanism for classifying and managing IP network traffic in order to provide QoS. It's common usage is to provide low-latency to critical traffic such as voice or another streaming media, while providing best-effort delivery for other, non-critical services such as web or file transfers.

DiffServ is defined in IETF RFC 2474 and later updated by RFC 3168 and RFC 3260 as an extension of well-known IP architecture based on packet routing and forwarding. On top of that, "the differentiated services architecture contains two main components. One is the fairly well-understood behavior in the forwarding path and the other is the more complex and still emerging background policy and allocation component that configures parameters used in the forwarding path." [17]

2.2 Resource Reservation Protocol

Another tool commonly used in several QoS approaches is the Resource Reservation Protocol (RSVP), defined by RFC 2205 [4]. This protocol has been designed for an Integrated services within the Internet. Before we will discuss Integrated Services (IntServ), we take a closer look onto RSVP, since IntServ relies on this protocol.

The main purpose of RSVP is, as stated in the protocol name already, to reserve resources for service, in order to avoid congestions, since the requested resources are fully dedicated for the service, so no one else on the network can use them.

RFC 2205 states that "RSVP provides receiver-initiated setup of resource reservations for multicast or unicast data flows, with good scaling and robustness properties." Further, "RSVP is also used by routers to deliver quality-of-service (QoS) requests to all nodes along the path(s) of the flows and to establish and maintain state to provide the requested service. RSVP requests will generally result in resources being reserved in each node along the data path." [4] RSVP is not a routing protocol, thus it can only work in cooperation with any configured routing protocol in the network.

2.3 Software Defined Networks

As could have been spotted in the previous section, SIP Single Port in combination with MPLS-TE separated the network between two logical layers - upper layer where the network "intelligence" and the decision making is located, and the bottom layer that is only responsible for forwarding the actual data.

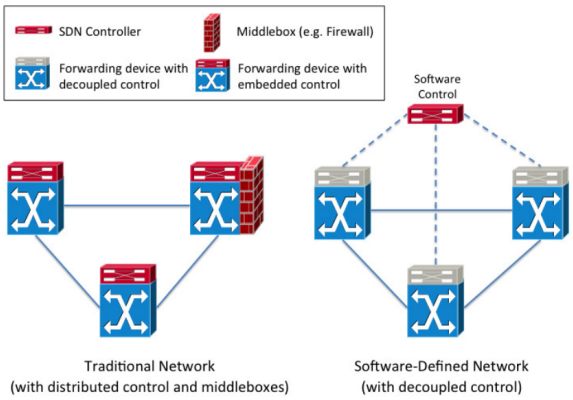

Figure 1: SDN architecture compared to conventional network. [18]

The conventional networks consisted of mostly closed and proprietary dedicated hardware devices with a stand-alone configuration and mostly the devices could only control their own behaviour, not considering what is good for the network as a whole entity. This resulted in the very exhausting network management, since the network configuration was distributed between practically all network devices, each of them having part of the overall network behaviour implemented. In such network, any on-demand or real-time changes are a non-trivial task, furthermore, making a configuration mistake in any of the network devices may led to overall network failure.

Software defined networking (SDN) is an approach to have the networks programmable in order to have the capacity to initialize, control, change and manage network behaviour dynamically via open interfaces. This approach emphasizes the role of software in the computer networks by abstracting the network for the data forwarding plane and the separate control plane. The layers and architecture of SDN is described in RFC 7426 [9] by IETF. This definition is very similar to what we have mentioned by SIP Single Port - the below layer, in SDN called data forwarding layer (DFL) is not capable of doing any logical decision - it is only responsible to deliver the actual data hop-by-hop via the network, while the upper layer, in SDN called control layer (CL) is a software component that controls the entire network from a logically centralized system. In this way, SDN makes it easier for network operators to evolve the network capabilities.

In the paper [18] author state that "the idea of programmable networks has recently re-gained considerable momentum thanks to the emergence of the Software-Defined Networking paradigm". SDN, often referred to as a "radical new idea in networking", promises to dramatically simplify network management and enable innovation through network programmability.

2.4 OpenFlow Protocol

Driven by the SDN principle of decoupling the control and data forwarding planes, OpenFlow [14] standardizes information exchange between the two planes.

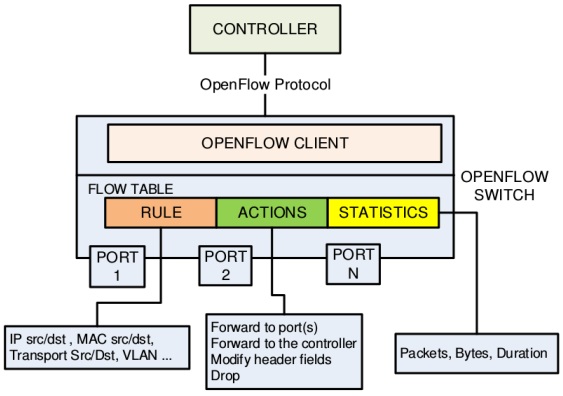

Figure 2: OpenFlow protocol architecture. [18]

In the OpenFlow architecture illustrated in Figure 2, the forwarding device, or OpenFlow switch, contains one or more flow tables and an abstraction layer that securely communicates with a controller via OpenFlow protocol. [18]

Upon a packet arrival at an OpenFlow switch, packet header fields are extracted and matched against the matching fields portion of the flow table entries. If a matching entry is found, the switch applies the appropriate set of instructions, or actions, associated with the matched flow entry. If the flow table look-up procedure does not result on a match, the action taken by the switch will depend on the instructions defined by the table-miss flow entry. Every flow table must contain a table-miss entry in order to handle table misses.

2.5 Unsplittable Flow Problem

At this point, the reader might consider applying the maximum flow algorithms like Ford-Fulkerson algorithm [6] to solve the optimization. When talking about jitter and delay sensitive data flows such as multimedia, we cannot perform the load balancing on a per-packet base. Instead, we must consider all such data flows unsplittable. Splitting these sessions will lead to a quality degradation due to different delivery times and therefore cause a jitter to rise or packets to be delivered out of order. [16]

2.6 Knapsack Problem

In general, the problem, where we want to distribute multiple data flows of given bandwidth between the multiple disjoint network paths of given capacity without exceeding the capacity of any path, can be expressed as a Knapsack Problem. According to [13], Knapsack is not a single specific problem, but a group of similar problems of the combinatorial optimization, where all af them belong to the class of NP-hard. We call these problems *the Knapsack problem modifications.*

The Knapsack Problem (KP) is the task of combinatorial optimization and is ranked as the task of integer linear programming. The Knapsack Problem has many applications in real life, and is therefore one of the most challenging issues in this area. For this reason, there have been many variations on this issue.

In the original KP, we have a set of items that all have some weight and some price (the weight and the price are not dependant on each other). Then, we have a knapsack of a certain loading capacity. The task is to select the subset of the items to be packed into the knapsack in order to carry the highest possible value without exceeding the knapsacks loading capacity. [5]

2.7 Online Knapsack Problem

The (classical) knapsack problem is given a set of items with profits and sizes, and the capacity value of a knapsack, to maximize the total profit of selected items in the knapsack satisfying the capacity constraint.

In the name Online Knapsack Problem the word ”online” means that items are given over time, i.e., after a decision of rejection or acceptance is made on the current item, the next item is given, and once an item is rejected or removed, it cannot be considered again. The goal of the online knapsack problem is the same as the offline version, i.e., to minimize or maximize the total cost. [10]

The authors in [10] further state, that by the observation: if all the items have the same size, then a simple greedy algorithm (called Lowest Cost First strategy) of picking items with the lowest cost first provides an optimal solution.

In another paper that deals with Online Knapsack Problem [3], the authors state that when the weights are uniform and equal to the weight constraint, the problem reduces to the famous secretary problem, or the problem of selecting online an element of maximum value in a randomly-ordered sequence. This problem was first introduced by Dynkin [21] in 1963.

2.8 Conversion Between Different Knapsack Problem Modifications

As mentioned above, some modifications of Knapsack Problems are very similar. So for example all solving methods for 0-1 Knapsack Problem are also applicable to some other Knapsack Problems. Most resources are devoted to 0-1 Knapsack Problem. The main reason is probably the possibility of transforming different Knapsack Problems into equivalent 0-1 Knapsack Problem form with only a generally limited increase in the number of variables. So it is possible to effectively solve this problem by using algorithms for 0-1 Knapsack Problems. [13]

Value independent Knapsack Problem can also be solved by any method for 0-1 Knapsack Problem, since we can simply transform it into 0-1 Knapsack Problem, so that $p_i = w_i$. However, in this case, it is much better to deal with it by specialized methods because they usually provide better results [13].

2.9 Brute Force Method

Brute force can be used to solve all the mentioned modifications of Knapsack Problem. It is a solution that simply examines all solution possibilities and chooses the best.

With 0-1 Knapsack Problem this algorithm has a complexity of $O(2^n)$, which is the number of all combinations. Obviously, with more complex variations of the problem, e.g. with more knapsacks, the number of combinations is even greater. Because of this complexity, this algorithm can only be used for small instances of the problem, and even then it is still slow and inefficient.

2.10 Branch & Bound Method

This method is an improvement of the brute force method. The principle of the method of branches and boundaries is to systematically go through all the potential solutions represented by the tree.

The algorithm passes through the branches of this tree that represent subsets of the set of solutions. Before an algorithm is embedded in a branch, it is always checked whether it is necessary to scan this branch at all. This is done by comparing the current branch with the lower and upper bounds of the optimal solution. If a branch can no longer deliver a better solution than the best solution found so far, it simply stops searching. The algorithm relies on an effective lower and upper bound estimate. By this method we can achieve an optimal solution, but not necessarily. It depends precisely on the way of calculating lower and upper bounds [22].

The complexity of this solution in the worst case is the same as in the brute force method, $O(2^n)$. However, in practice, the speed of this solution is significantly lower [24].

The Branch & Bound method is probably the most used and most effective method for obtaining an accurate solution for Knapsack Problems in general. It can be used in all the variations of the problem mentioned in this work. Of course, for each problem variation, boundaries are calculated differently, and although it provides significantly lower time than the brute force method, this time is still too long for this method to be used for real-time calculations.

2.11 Greedy Algorithms

These algorithms rely on a fact that in each step they select a local optimal solution in the hope that this solution will also be globally optimal. Heuristics are used to select a local optimal solution. Due to the combination of hope and heuristics, they do not need to always find the optimal solution.

It is a very simple and fast solution that consists of two phases. In the first phase, we sort the objects downward by price / weight ratio $\frac{p_i}{w_i}$. In the second phase, the objects are gradually stored in the knapsack until they exceed the capacity of the knapsack.

Thus, the most challenging part of this algorithm is the sorting of the elements that have minimal complexity of $O(n * log(n))$, for example, using *quicksort*. Inserting into the knapsack has a linear complexity. Thus, it is clear that this algorithm is much faster than the exact algorithms mentioned so far, but it will not always bring us the optimal solution. However, as the test results shown in [15] [2], the average deviation from the optimal solution decreases rapidly as the set of objects increases.

Therefore, this heuristics is useful in larger instances of the problem where the deviation is not so significant and exact methods would be too time consuming.

3. Value Independent Multiple Knapsack Problem

In this Section we introduce new Knapsack Problem modification, that combines two problem modifications mentioned earlier - Value Independent Knapsack Problem, where the objects do not have its weight and its value, but the value is equal to their weight, and the aim is to carry as much weight as possible, and the Multiple Knapsack Problem, where multiple containers (knapsacks) exist with each having its own capacity. The task remains bivalent - we either put the object into any of the knapsacks or we do not. Each object can be put into knapsack only once.

The mathematical model can be displayed by assuming that $p_i = w_i$ in the mathematical model of Multiple Knapsack Problem, as follows:

maximization:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} w_j x_{ij} \tag{1}$$

when:

$$\sum_{j=1}^{n} w_j x_{ij} \leq c_i, i \in M = \{1, ..., m\} \tag{2}$$

$$\sum_{i=1}^{m} x_{ij} \leq 1, i \in N = \{1, ..., n\} \tag{3}$$

$$x_{ij} \in 0, 1, i \in M, j \in N \tag{4}$$

$$w_j, c_j \in N; i, j, m, n \in N^0 \tag{5}$$

where:

x_{ij} is a binary variable expressing whether the object j has been inserted into the knapsack i

w_j is the ratio of $j-$object to knapsack capacity

c_i is the capacity of the $i-$knapsack

m is the amount of knapsacks

n is the amount of objects

3.1 Problem Description Using Value Independent Multiple Knapsack Problem

Let L be a set of all multimedia clients:

$$L = \{l_i : i \in \{1, 2, 3, ..., m\}\} \tag{6}$$

Let P be a set of all disjunct paths in the network:

$$P = \{p_k : k \in \{1, 2, 3, ..., r\}\} \tag{7}$$

Let ε be a function assigning 1 if a multimedia client is assigned to an path, otherwise 0, where:

$$\varepsilon : L \times P \rightarrow \{0, 1\} \tag{8}$$

All clients of the set L are interpreted as items. Paths of the set P are interpreted as knapsacks. The weight of item l, w_l, is the required bandwidth of a client and the size of a knapsack p, c_p, is the bandwidth capacity of the path.

For this problem modification we bring an approach made by us we call Quick Forward Lookup Algorithm.

4. Quick Forward Lookup Algorithm

Quick Forward Lookup Algorithm (QFLA) has been developed to address the problem of assigning clients to paths, in managing multimedia sessions. Based on the interpretation as Value Independent Multiple Knapsack Problem, where clients are objects and paths are knapsacks. QFLA as well as greedy algorithms does not guarantee finding the best solution.

The algorithm works on a model of gradual adding of items, and thus objects cannot be pre-arranged. In this case, it is an optimization algorithm that allows you to exchange items between different knapsacks. It allows us to increase the free capacity of some knapsacks so that another item can be placed into them, which otherwise would not be placed into any knapsack.

This algorithm has been first introduced as a concept in [7], where we provided proof of concept, the formal verification as well as some deeper research were still missing.

4.1 Algorithm Design

For the purpose of the algorithm, we define a knapsack problem UFKP (Unsplittable Flow Knapsack Problem), an instance of Value Independent Multiple Knapsack Problem, as follows:

$$x_{lp} \begin{cases} 1, & \text{if a client } l \text{ is assigned to a path } p \\ 0, & \text{if a client } l \text{ is not assigned to a path } p \end{cases} \tag{9}$$

In a formulation of this problem, we want to maximize:

$$\sum_{p=1}^{n} \sum_{l=1}^{m} w_l x_{lp}, \tag{10}$$

where n is a number of paths, m is a number of clients, x_{lp} is defined in (9) and w_l is the bitrate of all client's multimedia sessions.
While trying to maximize (10) following conditions have to be satisfied:

$$\sum_{l=1}^{m} w_l x_{lp} \leq c_p, \forall p \in \{1, ..., n\}, \tag{11}$$

$$\sum_{p=1}^{n} x_{lp} = \{0, 1\}, \forall l \in \{1, ..., m\}, \tag{12}$$

$$x_{lp} = 0 \text{ or } 1, l \in \{1, , m\}, p \in \{0, , n\}. \tag{13}$$

where c_p is a capacity of a path and x_{lp} is defined in (9).

At this point we would like to point to the situation where a client l has no multimedia sessions. In such situation w_l is equal to zero and therefore client l is not a subject to the UFKP.

There are several possibilities to solve clients that are not assigned to any path after an optimization process and their w is nonzero. Client's sessions either can be disconnected from the network with notification about the possible congestion in the network or client can be assigned to a path even though this will decrease its quality.

5. Algorithm Evaluation

In this Section we will evaluate algorithm formally as well as by testing its performance compared to known Knapsak Problem approaches.

5.1 Formal Verification of QFLA

The QFLA has not yet been formally verified. In the work [19] we prove that this algorithm is correct using direct proof.

Let m be the number of given knapsacks, R_i be the set of items assigned to knapsack i, R_{ia} be item a assigned to knapsack i, c_i be the capacity of knapsack i and we are supposed to assign given item with weight w. For every given R_i the capacity constraint $\sum_{a=1}^{|R_i|} R_{ia} \leq c_i$ must be satisfied.

$$\sum_{j=1}^{|R_i|} R_{ij} \leq c_i \tag{14}$$

must be satisfied. [19]

To prove the correctness of this algorithm we have to prove that the algorithm does not violate the capacity constraint, neither changes the overall number of assigned items otherwise than increasing it by one. We can interpret this statement as following theorem: [19]

Theorem A:

$$\sum_{k=1}^{m} |R_k| \leq \sum_{k=1}^{m} |R'_k| \leq \sum_{k=1}^{m} |R_k| + 1 \bigwedge \forall i \in 1, 2, 3, ..., m, \sum_{a=1}^{|R'_i|} R'_{ia} \leq c_i \tag{15}$$

where R' represents the sets after the assignment.

The algorithm consists of 3 stages, each of them represented by a lemma, in which k_l represents weight of items moved to knapsack l and x represents the knapsack where the item is supposed to be added: [19]

Lemma 1: In the first stage, the greedy knapsack algorithm stage, the item may only be assigned to a knapsack, but only if:

$$\sum_{a=1}^{|R_i|} R_{ia} + w \leq s_i \tag{16}$$

Lemma 2: In the second stage an item may be selected to be moved from one knapsack to another, but only if:

$$\sum_{a=1}^{|R_x|} R_{xa} + w - k_y \leq s_x \bigwedge \sum R_y + k_y \leq s_y \tag{17}$$

Lemma 3: In the last stage multiple items from one knapsack may be selected to moved from one knapsack to one or many other knapsacks, but only if:

$$\sum_{a=1}^{m} R_{xa} + w - \sum_{l=1}^{m} k_l \leq s_x \bigwedge \forall y \in \{1, 2, ..., n\} - i \tag{18}$$

$$\sum_{b=1}^{|R_y|} R_{yb} + k_y \leq s_y \tag{19}$$

Therefore we can conclude:

$$Lemma\,1 \bigwedge Lemma\,2 \bigwedge Lemma\,3 \implies Theorem\,A$$

Table 1: Utilisation of Knapsacks and Required Calculation Time on Different Approaches [12]

	util. [%]	time [ms]
Greedy algorithm	99.93%	0.1 ms
Branch & Bound	100%	1165.92 ms
QFLAv1	99.81%	0.4 ms

and so prove that Quick Forward Lookup Algorithm is correct as for we proved that no item has been removed from knapsacks, only on may be added and that all capacity constraints are satisfied. [19]

5.2 Algorithm Performance Evaluation

In this test, all evaluated algorithms have been tested for a set of 100 items of different sizes, on top of three knapsacks with different, but predetermined sizes (6000, 4000 and 10000).

For items, we used data from dataset [1]. This resource provides a large amount of data to test 0-1 Knapsack Problem in csv format. However, since we are dealing with a modification of the 0-1 Knapsack Problem, we can use this data for any other Knapsack Problem. The sizes of the items ranged from 1 to 1000. The price of the item was not considered.

We ran all tests repeatedly 5 times (with random objects from datasets) and averaged the results we got (Table 1).

When testing the Branch & Bound algorithm, the algorithm was limited to only one backtrack. In the table we can see that the branch & bound algorithm has found the maximum solution (total sum of all knapsacks - 20000).

The more detailed test results presented in [12] show that the calculation time does not change noticeably for larger instances and we can see that larger instances achieve better results. It was also unusual for the one case the Branch and Bound algorithm found a solution with a profit of only 19803, in other cases the final profit of this algorithm was repeatedly 20000. This solution is the only one of all the solutions worse than the greedy algorithm. As far as the calculation times are concerned, the times are clearly increasing exponentially with a larger number of objects.

Branch & Bound, which has the longest solution times, clearly stands out from the compared algorithms. It has the best results on average, but in real time applications, we require different algorithm.

The greedy algorithm and QFLA had quite similar results in testing, but the greedy algorithm has always found a slightly better solution in a shorter average time. The only, but in our case very important advantage of QFLA, is that it takes into account also the amount of needed reallocations, and it tries to find the solution in the way that minimises number of reallocations will be needed.

6. Conclusion

We described the basic principles of multimedia sessions, mechanisms on quality of service in IP networks, and the principles and advantages of the SDN approach. We described existing SDN protocols and controllers, and we mentioned the related projects that deal with the problems of quality of multimedia services. Furhter we described the problems with load balancing, when the per-packet load balancing approach is useless and we consider whole data flow unsplittable.

We explained the problem of distributing these flows of given required bandwidth between the disjoint network paths of given capacity using the well known Knapsack Problem, where we brought up our own modification of this problem we call Value Independent Multiple Knapsack Problem. We described the known methods on solving various modifications of Knapsack Problem, such as brute force, branch and bound, dynamic programming or greedy algorithms.

There exist several approaches on how to solve the problems related to Online Value Independent Multiple Knapsack Problem, all of the however do not take into account the number of steps required to achieve the improved items-to-knapsack distribution. Since in the real network every step equals a configuration change that may affect QoS, one the the requirements was that the algorithm must look for the improved solution with minimum reallocations (configuration changes).

For this purpose, we took the prototype of QFLA algorithm which has been designed by us in [7], that in case, when new client (flow) cannot be inserted directly into any of the paths, tries to move existing flows to different paths in order to relax one path enough for the new flow to be inserted without exceeding the capacity of the path.

The most important areas for the future work that can be done this project are following: In all the scenarios, we were only dealing with knapsacks and with items, without any other constrain. That means, any of the items can be packed into any of the knapsack if there is free enough capacity (the size matters only). Yet remains unsolved, how would be the solution proposed in this thesis be suitable for the scenarious where the items would have not only their size, but also their price, thus we would need to consider some priorities of packing certain item into the knapsack with higher importance than some other.

Acknowledgements. This research was supported by the Ministry of Education, Science, Research and Sport of the Slovak Republic, Incentives for Research and Development, Grant No.: 2018/14427:1-26C0. The work was also supported by the Operational Program Research and Innovation for the project: Research of advanced methods of intelligent information processing (ITMS code: NFP313010T570), co-financed by the European Regional Development Fund, and the Research and Development Operational Program for the project: Support of Center of Excellence for Smart Technologies, Systems and Services II (ITMS 26240120029), co-financed by the European Regional Development Fund. It was also partially supported by the grants APVV-15-0731, KEGA 011STU-4/2017, and VEGA 1/0836/16.

References

[1] Pisinger, d.: Small coefficients. Accessed Online: http://www.diku.dk/ pisinger/smallcoeff_pisinger.tgz, 2019.

[2] D. Antonín. Řešení problému batohu metodou hrubé síly a jednoduchou heuristikou, 2013. Accessed Online: http://data.antonindanek.cz/paa-du1.pdf.

[3] M. Babaioff, N. Immorlica, D. Kempe, and R. Kleinberg. A knapsack secretary problem with applications. In M. Charikar, K. Jansen, O. Reingold, and J. D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 16–28, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[4] B. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. Resource reservation protocol (rsvp) – version 1 functional specification. RFC 2205, RFC Editor, September 1997. http://www.rfc-editor.org/rfc/rfc2205.txt.

[5] E. Šemnická. Praktické řešení úlohy batohu [online], 2008 [cit. 2019-07-08].

[6] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[7] M. Galinski. Multimedia sessions optimization using sip single port. Master's thesis, Slovak University of Technology in Bratislava, 2016.

[8] M. Garcia-Martin. Rfc 4083: Input 3rd-generation partnership project (3gpp) release 5 requirements on the session initiation protocol (sip). *Internet Engineering Task Force*, 2005.

[9] E. Haleplidis, K. Pentikousis, S. Denazis, J. H. Salim, D. Meyer, and O. Koufopavlou. Software-defined networking (sdn): Layers and architecture terminology. RFC 7426, RFC Editor, January 2015. http://www.rfc-editor.org/rfc/rfc7426.txt.

[10] X. Han and K. Makino. Online minimization knapsack problem. In E. Bampis and K. Jansen, editors, *Approximation and Online Algorithms*, pages 182–193, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[11] K. T. B. A. M. T. Hilmi E. Egilmez, S. Tahsin Dane. Openqos: An openflow controller design for multimedia delivery with end-to-end quality of service over software-defined networks. *Koc University Istanbul Turkey*.

[12] P. Hlavaty. Knapsack problem, 2018. Slovak University of Technology in Bratislava, Bachelor's Thesis, Supervisor: M. Galinski.

[13] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Inc., New York, NY, USA, 1990.

[14] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, Mar. 2008.

[15] T. Michal. Problém batohu, 2009. Accessed Online: https://woq.nipax.cz/sobaka/01_problem_batohu.php.

[16] J. Muranyi. *Optimization of Multimedia Flows in Multipath Networks*. PhD thesis, Slovak University of Technology in Bratislava, 2015.

[17] K. Nichols, S. Blake, F. Baker, and D. L. Black. Definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers. RFC 2474, RFC Editor, December 1998. http://www.rfc-editor.org/rfc/rfc2474.txt.

[18] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti. A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials*, 16(3):1617–1634, 2014.

[19] F. Pavkovcek. Load balancing as a modified knapsack problem. Master's thesis, Slovak University of Technology in Bratislava. Supervisor: M. Galinski, In progress, not published yet.

[20] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. Sip: Session initiation protocol. RFC 3261, RFC Editor, June 2002. http://www.rfc-editor.org/rfc/rfc3261.txt.

[21] S. S. Seiden. An optimal online algorithm for bounded space variable-sized bin packing. In *International Colloquium on Automata, Languages, and Programming*, pages 283–295. Springer, 2000.

[22] A. Shaheen and A. Sleit. Comparing between different approaches to solve the 0/1 knapsack problem. *International Journal of Network Security*, 16:1–10, 07 2016.

[23] S. S.Kandula, D.Katabi and A.Berger. Dynamic load balancing without packet reordering. *ACM SIGCOMM Computer Communication Review*, 37(2):51–62, 2007.

[24] K. Tomáš. 3. úloha - problém batohu ii., 2005. Accessed Online: https://woq.nipax.cz/sobaka/01_problem_batohu.php.

Selected Papers by the Author

M. Galinski, T. Vrtal and I. Kotuliak, "Network Controller Extension for Unsplittable Data Flows in the SDN Environment," 2019 17th Int. Conf. on Emerging eLearning Technologies and Applications (ICETA), 2019, pp. 197-203.

M. Galinski, J. Volko, I. Kotuliak, "Binary Search Tree as an approach on solving Modified Knapsack Problem", Sborník příspevku PAD 2019 – elektronická verze PAD2019, 2019, pp. 39-42.

L. Mastilak, M. Galinski, I. Kotuliak and M. Ries, "Improved Smart Gateway in IoT," 2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA), 2018, pp. 349-354.

Multi-Chain Architecture for Blockchain Networks

Kristián Košťál*
Institute of Computer Engineering and Applied Informatics
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 2, 842 16 Bratislava, Slovakia
kristian.kostal@stuba.sk

Abstract

This paper focuses on interoperability between blockchain networks, whether existing or new. The issue of interoperability is a very current topic and is the subject of research by many articles or institutions. The current trend in the articles is to address interoperability only between homogeneous networks, i.e. the so-called Sharding, and not at all dealing with interoperability between different networks. The second trend is to focus on two specific blockchain networks and interconnect them as efficiently as possible. However, both of these trends are flawed, as a large number of new heterogeneous networks have recently emerged, which are incompatible with each other, so that all processes take place in isolation. If they want to communicate with another network, this is almost impossible because there is no standard for interlock blockchain communication. The only option is to exchange assets within the exchange, but this is only a scenario for cryptocurrencies and not for other use cases. Based on the research, an architecture is proposed that uses an API to connect existing blockchain networks and allows communication between any blockchain. From this blockchain it is possible to derive other blockchains that are scalable. The architecture has been implemented and experimental verification has shown scalability up to 18,000 transactions per second.

Categories and Subject Descriptors

C.2.1 [**Networks**]: Network Architecture and Design—*distributed networks*; C.2.4 [**Networks**]: Distributed Systems—*distributed applications, distributed databases*; D.2.12 [**Software Engineering**]: Interoperability; H.2.4 [**Database Management**]: Systems—*distributed databases*

*Recommended by thesis supervisor: Prof. Ivan Kotuliak Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on September 22, 2020.

Košťál, K. Multi-Chain Architecture for Blockchain Networks. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 12, No. 2 (2020) 8-14

Keywords

blockchain, cross-chain, multi-chain, atomic swaps, interoperability

1. Introduction

We live in a digital age. The Internet is used for everyday communication, bill payments, shopping, charity support or just for fun. Among them, blockchain technology creates another digital dimension and offers new business opportunities.

A key feature is storing data and transactions that change it in block strings. It acts as a distributed database that processes an ever-growing list of transactions for which no authority owns the data. Thanks to this function, the blockchain is extremely resistant to unauthorized modifications to the stored data. The financial sector was therefore the largest supporter. Publicly, the most perceived blockchain applications are cryptocurrencies [8, 24]. Even some of the most dominant regulators in the United States, such as Commodity Futures Trading Commission (CFTC) and Securities and Exchange Commission (SEC), see the positive potential of this technology.

In addition to its use in cryptocurrencies transactions, the benefits of blockchains, such as decentralized information verification and tamper resistance, have been noticed by various sectors [20, 21, 10]. Important applications include the value register, the valuable website and value ecosystem services. Related application industries include logistics, financial systems, medical records, data collection and verification in Internet of Things (IoT), supply chain management, stock or option trading, social networking software, electronic patient records, micropayments / mobile payment systems , asset transactions and distribution of digital products. People hope that blockchains will be able to play the role of trusted machines in the operation of these systems. Keeping a detailed record of related information and resolving information asymmetry problems (where one party to an economic transaction owns more related information than the other party) will allow a reliable record to be created. In the cases used above, a large amount of information will need to be recorded on the blockchain [14].

We will need more robust blockchain networks than are used today to store large amounts of data. This robustness will mainly face problems with scalability, speed of transactions and also privacy challenges [28, 25]. The problem is also in the blockchain community, as there are many more initiatives aimed at creating new blockchains

or cryptocurrencies than at developing possible connections and means of communication between existing blockchain networks. For such a solution we need interoperability [11, 22] between today's blockchains and that is the main focus of this work.

2. State of the Art

Blockchain is, simply put, the technology of distributed ledger book (DLT). Blockchain is a decentralized distributed database, which is stored as a chain of blocks, that are cryptographically ordered. The blocks contain transactions between addresses of users. All of this happens in a peer-to-peer network. There exist more kinds of blockchain technology: public, private, consortium. Furthermore, these can be divided by two groups: permissionless, permissioned. When we talk about interoperability, we think communication between two distinct blockchains by any means. Therefore, we need to analyze the options and techniques to achieve inter-operable ledgers.

2.1 Cross-chain

We can understand cross-chain technology as a bridge connecting more blockchains. Its main application is the implementation of atomic cross-chain swaps between different blockchains, asset conversion or inter-block communication [4]. There are obvious obstacles to the distribution of value between blockchains. The cross-chain is a complex distributed coordination task. It needs not only a separate authentication capability for blockchain nodes, but also decentralized input, as well as data acquisition and authentication beyond the blockchain [26]. Cross-chain technologies currently include [3]:

- **Notary schemes** - one side decides to take any action on chain B when a specific event on chain A occurs. Means that a group of trusted parties comes to Byzantine-Fault Toleracne (BFT) consensus over the state of the ledger according to some action on another ledger. The notaries usually exist on separate ledger that they manage to ensure interoperability between two or more ledgers [16]. This is technologically one of the simplest solution to accomplish interoperability goal under its trust model. The model assumes that less than some fraction of the notaries act in Byzantine way [3];
- **Sidechains / Relays** - one chain is capable of reading and validating events on other chains. Technique that allows chains to act based on events from another ledger with the help of a smart contract. They are a more direct approach for chain interoperability without trusted intermediate participants as opposed to notary schemes [3]. They usually need copies of block headers from the other chain. These block headers usually contain Merkle Root tree hash by which any transaction can be proven to be contained in the block from the other chain [16]. Relay schemes are quite powerful because they allow for asset portability, atomic swaps or other more complex use case without restrictions [3]. The drawback is that the underlying blockchains one in case of one-way and both in case of two-way relays needs to support such smart contracts;
- **Hash-locking** - transactions on two chains have the same activation, usually a disclosure of some value. A technique where operations on two interoperating chains are set up in a way that they have the same trigger which is usually a revelation of the pre-image of a particular hash [3]. Hash-locking is simpler compared to relay schemes in which some copy of one ledger must be stored on another ledger because hash-locking only require an exchange of a single hash between two ledgers thus requires significant less exchange of data to achieve interoperability. The primary focus of hash-locking is to allow atomic swaps between ledgers without requiring a third party as it is with notary schemes [16]. The key drawback of hash-locking is that the underlying blockchain must support a special type of smart contracts, called Hash-TimeLock Contracts.

These are three main techniques. Moreover, we can define 4 potential use-cases for interoperability of blockchains as laid out in [3] and [16]:

- **asset portability** - transfer asset from one blockchain to another;
- **atomic swap** - swap asset between two blockchains in atomic way;
- **cross-chain oracles** - are used to read data from outside of blockchain so it can be act upon them;
- **cross-chain asset encumbrance** - lock up assets within one ledger based on locking conditions dependent on another ledger.

The techniques and use-cases are summarized in Table 1.

2.2 Related Work

There exist more solutions which try to make interoperabilty between blockchains. The nowadays state of the art solutions in this research area, according to my analysis and best knowledge, are Cosmos [17] and Polkadot [27]. They are both trying to solve similar problems, concretely introducing interconnection between any blockchain network with the help of centralized Cosmos's Hub (or Polkadot's Relay Chain). To the best of my knowledge they are now more focusing on homogeneous (similar in format of contracts, accounts and consensus) blockchains based on their own engines (Tendermint in Cosmos and Substrate[1] in Polkadot).

Other solutions are also suffering the same issue and a lot of more ones. Blocknet [6] focuses only on solving problems with cryptocurrency assets, concretely making a fully decentralized exchange. BTC Relay [5] concentrates only on one kind of interconnection, specifically connecting Ethereum and Bitcoin blockchains. Corda [12] is not blockchain in its fundamentals. It has some attributes of blockchain technology but tries to solve all problems only with notary schemes. Even if Corda was blockchain, it would not be able to make a connection between existing blockchain networks because Corda needs to have specifically implemented systems to be able to communicate with each other. InfiniteChain [14] introduces only sidechains to one blockchain network to make

[1] a framework for building decentralised systems; `https://www.parity.io/substrate/`

Table 1: Cross-chain Techniques and Their Use Cases

	Hash-locking	Notary schemes	Sidechains / Relays
Atomic swaps	Yes	Yes	Yes (only 2-way)
Asset portability	No	Yes (requires long-term notary trust)	Yes
Cross-chain oracles	No	Yes	Yes
Asset encumbrance	No	Yes (requires long-term notary trust)	Yes

it more private, faster, scalable and distribute high data volumes. Thus it cannot connect more existing blockchain networks. Interledger Protocol (ILP) [13] is a protocol for communication between any ledgers, not only oriented into blockchains but also to ledgers held by existing banks. However, as the ILP uses Hash Timed Locks, it mandates that all sides of communication implement contracts. The Hyperledger Quilt [15], side by side with Ripple, is just a real implementation of Interledger protocol into blockchain technology, so it also shares the same drawbacks as ILP. Lightning Network [23] is comparable to InfiniteChain but it is not a sidechain. A sidechain counts on its own blockchain. This network is, with the help of a two-way peg, coupled to the existing blockchain. On the other hand, the Lightning network consists of native Bitcoin 2-of-2 multi-sig transactions. The cross-chain communication with Lightning Network works as long as all the chains support the same hash function to use for the hash lock. Nodes cooperating in the Lightning Network have to be always online in order to send and receive payments, and this lowers the security as cold wallets cannot be used. Rootstock [18] is the next example of sidechain, but it has a significant disadvantage, that it is suited only for Bitcoin blockchain. Despite that, its only use is to make faster transactions with the help of Simplified Payment Verification[2]. TAST project [1] is still in development and not finished, but their approach with tokens in all blockchains requires that each blockchain has smart contracts functionality to create the tokens. Moreover, in order to function, all wallet balances and movable tokens must be in all participating blockchains. This can radically higher storage demands. Wanchain [19], with the help of Ethereum smart contracts, makes an interconnection between different Decentralized Apps (DApps), but for now, it works only with Ethereum-based blockchains. XClaim [31] is not a universal solution for blockchain interoperability. By far, it was implemented only between Bitcoin and Ethereum. The drawbacks are the need to have at least one smart contracts compatible blockchain, for each pair of blockchains you need to implement specific smart contract only for that pair, and the whole trust in transactions is based on a vault, which locks the funds.

2.3 Summary

A number of competing projects presented here aimed at creating a unified platform for inter-blockchain communication appeared. However, despite the large number of use cases and attempts to address them, the underlying problem of interblockchain communication has not been clearly defined, nor have the related challenges or the existing research [29] been studied. The lack of generality and dependence on the specific implementation of blockchains are disadvantages known to the vast majority of existing solutions. All blockchain solutions work on their "sand" and are not interested in communication between a huge number of blockchains and the interconnection of existing solutions. None of the options described above support plug-in connection of existing blockchains. Only some of the solutions may comply with some government regulations or may communicate with standard databases. In addition, some of the designs add additional overheads to already slow and unpredictable blockchain networks, which only limits functionality. So there are many gaps in interoperability that need to be addressed.

3. Research Goals

According to the references and analyzed solutions, we certainly see room for improved interoperability between existing blockchain networks not only specialized in cryptocurrencies, but also databases or perhaps government and state technologies.

All our research is focused on the use of modern blockchain techniques and related technologies in combination with our knowledge of networking and architecture. Summary of the main goals:

- Introduce a new **multi-chain blockchain architecture** to connect two or more different blockchains to allow interoperability between them.
- Enable **attach existing blockchains** in an almost plug & play manner without the need to implement specific requirements on their part.
- Introduce **real-world asset backing mechanism** in token-driven blockchain, e.g. a gold-protected token as a store of value to prevent volatility.

In all three theses, it is important to think about security and, where necessary, ensure that assets can be exchanged between blockchains (for example, in the case of cryptocurrency volatility, redeeming an asset from the real world with a more stable value) and ensuring that this problem can be solved without a centralized third party, e.g. an exchange. The standard approach to atomic swaps

[2] `https://en.bitcoinwiki.org/wiki/Simplified_Payment_Verification`

between blockchain chains is currently hashed time-locked contracts and smart contracts, which function as two-way relays.

4. Architecture Design

In the proposal to solve the described problems we have two parts. The first is a unique double blockchain technology, where one is public and contains regular transactions, and the second is private, which stores information about the real coverage of the assets from the first blockchain, such as gold. The second part of the design is in the whole network ecosystem for new blockchains and also for existing ones. We decided to think of the blockchain as a color, so each blockchain has its own color assigned to it. That's why we present the color blockchain. The network is controlled by colors grouped into color spaces. The network topology is logically structured as a network of clearly colored networks, as shown in the Figure 1.

4.1 Double-blockchain

For double blockchain purposes, we will use authorized blockchains, authentication will only be possible through the authentication block, and the token register in the authentication subsystem will remain private. The validation block is the connection point between the public and private blockchain. The authentication block can be an application interface (API) or a smart contract. This option represents the detail of the implementation, both cases should work without problems, but the API is an option that can be implemented faster. In order to be able to verify the token validity status in a trouble-free manner, i.e. validation will take place after the transaction is signed without significant delay in the speed of the transaction, we will need a fast scalable consensus algorithm. The ideal consensus algorithm seems to be the Practical Byzantine-Fault Tolerance (PBFT) used in Hyperledger Fabric, which can reach more than 20,000 transactions per second (TPS) [9]. However, to complement the results of the mentioned article, their testing was performed on Fabric 1.2 and only with a critical path, not with complete nodes. According to the authors, the results will be about 25-30 % lower for complete nodes. Double-blockchain is an exact solution to my goal number three.

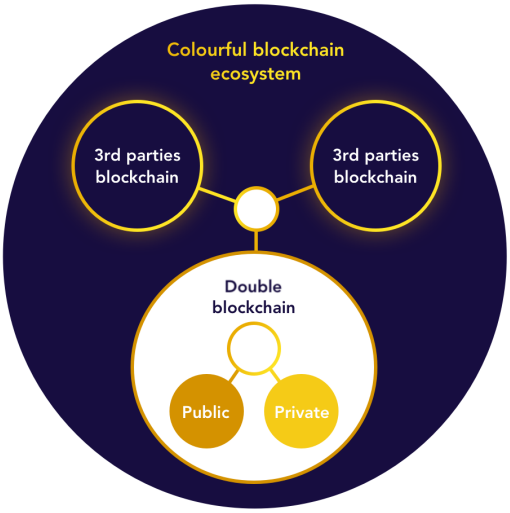

Figure 1: Color blockchain network.

4.2 Color Blockchain

An architecture is proposed to solve interoperability problems between current blockchain networks. The proposed design introduces a multi-chain architecture for new, as well as existing blockchains. We called the architecture color blockchain. We do not provide any details or instructions regarding implementation, they are up to the developers. Examples of such details are the use of Merkle trees, specific consensus algorithms, cryptography, keys, and other blockchain bases that are inherited from the blockchains on which applications are built. Some basic concepts of architecture are given here.

4.2.1 Problem with "Too Large" Data

The coexistence of color blockchains can lead to a huge number of generated blocks during the life of the network. To relieve the pressure on the required performance and storage, we need the Epoch concept based on light clients [30] and Merklix tree[3].

4.2.2 Participation in Colors and Color Space

Each block belongs to exactly one color. This color belongs to the color space, and there is a tree hierarchy of color spaces.
Nodes participating in a particular color usually process only blocks of that color. They can (but are not obliged - depending on the rules in the color space) also process blocks belonging to the nearest / smallest color space to which they belong. Zero color blocks in color spaces are mainly used for color space management and inter-colored communication.

4.2.3 Color Blockchain Custom Network

The color blockchain network is a complete blockchain technology that can handle thousands of individual blockchains and a wide range of configurations. As part of the introduction, the network will create two custom blockchain implementations on the network: Color Blockchain and Color Verification Double Blockchain. Both of these blockchains can be realized in separate color spaces as individual colors. This color space can have a special customized set of rules that validates the properties of tokens in one color using properties in another color.

4.2.4 Smart Smart Contracts

Smart contract design allows to use color blockchain as a robust execution environment with pluggable environments and contract configuration. Smart Contract Execution Engine isolates the execution of smart contracts in a quarantine environment, and therefore protects the network from malicious behavior and increases security [7].

Finally, the key features of the architecture are:

- Blockchain colors - the color blockchain network is designed from the ground up based on the idea of partial blockchains. It can be divided into separate sub-blockchains with their own rules and behavior.
- Enormous readiness - the networks is designed to allow a huge number of operations in terms of throughput and overall capacity thanks to its concept of distributed processing and blockchain snapshots.

[3] https://www.deadalnix.me/2016/11/06/using-merklix-tree-to-shard-block-validation/

- Double-blockchain - in the center of color blockchain is the concept of double blockchain transaction verification. This can be applied to support mechanisms, e.g. gold-covered assets, cryptocurrency-covered assets, etc.

- Attaching existing blockchains - the only prerequisite for communicating with existing blockchains is the API. The API can be provided directly by the target blockchain, some API service for blockchains, or the developer can build it himself.

These are the key concepts of architecture. If they work, the whole architecture works like a monolith. Colors are the basic thing about an architecture that provides information about transactions, interoperability, but also makes the system easier to understand. The second feature concerns scalability. Slow blockchains already exist, but fast blockchains with a very high number of transactions per second are, as the overview shows, very current and academia and industry are trying to reach new milestones. Double-blockchain is a very innovative feature that shows other use cases where blockchain can be used. This technology can be selected for any combination of blockchain technologies (public, private). The combination of existing blockchains and distributed ledger technologies, i.e. Heterogeneous interoperability is the highest priority in this work and also a current research topic in the field of inter-blockchain communication.

5. Experimental Verification

We decided to implement a smaller grain of the proposed solution to see the key aspects of the architecture and to demonstrate the key concepts behind the design in a simple way.

5.1 Experimental Methodology

For experimental purposes, we implemented the topology shown in the Figure 2. The decision about colors and color spaces was random without special significance. We have grouped related blockchains with respect to their consensus algorithm used, but note that this is only current choice for experiments.

In this experimental topology, we use APIs provided from blockchains. However, if the blockchain does not provide its own API, we must use some service that the API will create or make available. For this service we use CryptoAPIs [4], which is an infrastructure layer that provides APIs, WebHooks and Web Sockets for some of the existing blockchain protocols, e.g. Litecoin, Bitcoin, Dash, Dogecoin and more.

The blockchains selected for testing were the public testing networks Bitcoin, Ethereum and Eos. These three existing blockchain networks are sufficient to demonstrate functionality and communication with existing blockchains. Color blockchain uses the Tendermint core [2].

All experiments were performed on Amazon EC2 instances of type *m5n.4xlarge*, where we ran the tests with 4, 8, 16 and 32 instances. Each was a validator for the core network and some were validators and vaults for existing blockchains and double-blockchain. We repeated each test 100 times and the results shown here are the median values.

[4] `https://cryptoapis.io/products/blockchain-apis/`

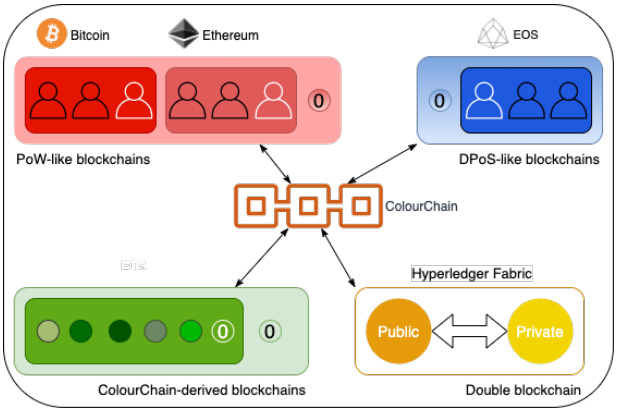

Figure 2: Experimental topology.

The scenario was to verify the speed of transactions within the color blockchain network. When we compare it with the Tendermint from which this blockchain is built, we get very similar results, almost 20,000 TPS. The results are shown in the Figure 3.

5.2 Summary

This chapter described the verification of the proposed architecture. The verification started with the setting up of networks and processes, which also served as a basic verification for defined transactions and blocks. Verification with experimental implementation followed. Both selected methods are independent of each other and confirm the functionality of parts of the proposed architecture. In the experimental case, we verified the design by implementing part of the architecture and the sample network using several different blockchains. We divided this case into two scenarios. One where we focused on double blockchain technology to point out its speed and minimal impact on extension when verifying gold support for each transaction. The test showed minimal differences in times with and without validation, the only case where the difference was more seen, i.e. 831 ms, was with 50,000 transactions, but so many transactions usually do not occur in one blockchain network at once. In the second scenario, we verified the partially implemented color blockchain architecture using three existing blockchain networks, a double-blockchain, and six color blockchain-derived blockchains. In addition, there were APIs to connect to existing blockchains. The results in this scenario show that our network is high-speed and its slowdown is only affected by the length of the verification time in the source and destination blockchain. Subsequently, we verified the scalability on blockchains derived from color blockchain, which are not significantly slowed down by block time, and here it turned out that the entire color blockchain network managed up to 18,000 TPS. During the network latency and throughput test scenarios, we monitored the impact of our implementation and use of consensus algorithms on network properties. Measurements were performed in a real geographically distributed environment to indicate network operation if deployed. The real environment was set so that its properties do not affect the quality. In all experiments, we verified that the color blockchain network has a minimal impact on the quality of services provided.

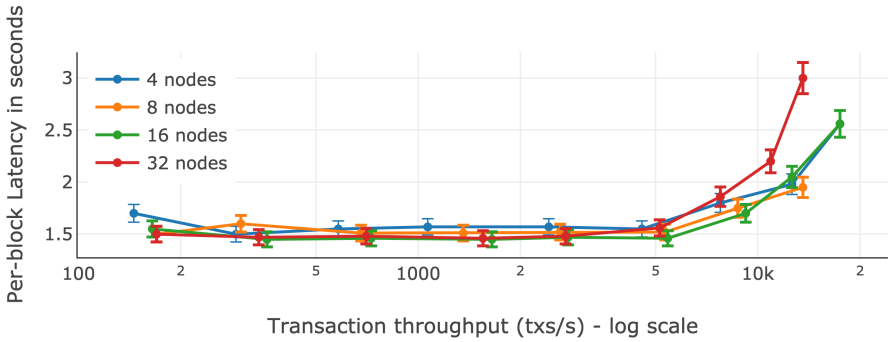

Figure 3: Delay vs. throughput in a testing environment.

6. Conclusion

In this work, we presented a solution to problems with the interoperability of various blockchain networks. Our architecture supports atomic swaps between existing blockchain networks using vaults and collateralization. There are also notary schemes. The current literature has shown that we cannot implement blockchain interoperability without a trusted third party [29]. However, in our solution, validators in a trusted third party come from participating blockchains, so even though our transmission chain is called "centralized," if we use nodes from existing networks where they are trusted, they will also be trusted in the color blockchain.

The results of verifications confirmed the functionality of the proposed architecture and the fulfillment of the defined goals. We introduced a new multi-chain blockchain architecture for interoperability that has proven itself, so the first goal is complete. The second goal is also achieved thanks to the API for existing blockchains and the introduction of colors and color spaces. We tested the connection of three different blockchains to our network; linking a new blockchain is a process that takes a few seconds, because the only requirement is to map the functions of the target blockchain to color blockchain. The third goal is achieved using double blockchain technology. In addition, atomic swaps are achieved in the architecture using smart contracts (i.e. relay schemes) and automatic locks between accounts via a transmission chain (i.e. notary scheme), but the system is also ready for hash locks, this will only be an implementation detail.

Acknowledgements. This research was supported by the Ministry of Education, Science, Research and Sport of the Slovak Republic, Incentives for Research and Development, Grant No.: 2018/14427:1-26C0. This publication was created thanks to support under the Operational Program Integrated infrastructure for the project: Research in the field of blockchain technology with connection to online payment services, ITMS 313022U641, co-financed by the European Regional Development Fund. It was also partially supported by the grants APVV-15-0731, KEGA 011STU-4/2017, and VEGA 1/0836/16. The author would like to thank for financial contribution from the STU Grant scheme for Support of Young Researchers.

References

[1] M. Borkowski, D. Mcdonald, C. Ritzer, and S. Schulte. Towards Atomic Cross-Chain Token Transfers: State of the Art and Open Questions within TAST. Technical report, TU Wien, Vienna, Austria, 2018.

[2] E. Buchman, J. Kwon, and Z. Milosevic. The latest gossip on BFT consensus. *arXiv*, 1807(4938):14, jul 2018.

[3] V. Buterin. Chain Interoperability. *R3 Research Paper*, 2016.

[4] Z.-d. CHEN, Z. YU, Z.-b. DUAN, and K. HU. Inter-Blockchain Communication. In *Proc. of the 2nd Int. Conf. on Computer Science and Technology (CST 2017)*, pages 448–454, Guilin, China, 2017. DEStech Publications.

[5] L. Coleman. BTC Relay Bridges BTC With Ethereum, Allowing BTC Verification For Smart Contracts, 2016.

[6] A. Culwick and D. Metcalf. The Blocknet Design Specification. Technical report, 2018.

[7] G. G. Dagher, C. L. Adhikari, and T. Enderson. Towards Secure Interoperability between Heterogeneous Blockchains using Smart Contracts. In *Future Technologies Conf. (FTC) 2017*, pages 73–81, Vancouver, Canada, 2017. Science and Information Conferences.

[8] I. Eyal. Blockchain Technology: Transforming Libertarian Cryptocurrency Dreams to Finance and Banking Realities. *Computer*, 50(9):38–49, 2017.

[9] C. Gorenflo, S. Lee, L. Golab, and S. Keshav. FastFabric: Scaling Hyperledger Fabric to 20,000 Transactions per Second. In *2019 IEEE Int. Conf. on Blockchain and Cryptocurrency (ICBC)*, pages 455–463. IEEE, 2019.

[10] Y. Guo and C. Liang. Blockchain application and outlook in the banking industry. *Financial Innovation*, 2(24):1–12, dec 2016.

[11] T. Hardjono, A. Lipton, and A. Pentland. Towards a Design Philosophy for Interoperable Blockchain Systems. *arXiv*, 1805(5934):27, may 2018.

[12] M. Hearn. Corda: A distributed ledger. Technical report, 2016.

[13] A. Hope-Bailie and S. Thomas. Interledger: Creating a Standard for Payments. In *Proc. of the 25th Int. Conf. Companion on World Wide Web - WWW '16 Companion*, pages 281–282, New York, New York, USA, 2016. ACM Press.

[14] G.-H. Hwang, P.-H. Chen, C.-H. Lu, C. Chiu, H.-C. Lin, and A.-J. Jheng. InfiniteChain: A Multi-chain Architecture with Distributed Auditing of Sidechains for Public Blockchains. In S. Chen et al., editors, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 10974 LNCS, pages 47–60. Springer International Publishing, Cham, jun 2018.

[15] Hyperledger Inc. Hyperledger Gets Cozy With Quilt, 2017.

[16] T. Koens and E. Poll. Assessing interoperability solutions for distributed ledgers. *Pervasive and Mobile Computing*, 59(101079):1–10, oct 2019.

[17] J. Kwon and E. Buchman. Cosmos: A Network of Distributed Ledgers. Technical report, 2018.

[18] S. Lerner. Drivechains, Sidechains and Hybrid 2-way Peg Designs. Technical report, RSK Labs Ltd., 2016.

[19] J. Lu, B. Yang, Z. Liang, Y. Zhang, S. Demmon, E. Swartz, and L. Lu. Wanchain: Building Super Financial Markets for the New Digital Economy. Technical report, 2017.

[20] M. Mettler. Blockchain technology in healthcare: The revolution starts here. In *2016 IEEE 18th Int. Conf. on e-Health Networking, Applications and Services (Healthcom)*, pages 1–3. IEEE, 2016.

[21] Q. K. Nguyen. Blockchain - A Financial Technology for Future Sustainable Development. In *2016 3rd Int. Conf. on Green Technology and Sustainable Development (GTSD)*, pages 51–54. IEEE, nov 2016.

[22] L. Pawczuk, J. M. Nielsen, P. K. H. Sin, and N. Hewett. Inclusive Deployment of Blockchain for Supply Chains: Part 6-A Framework for Blockchain Interoperability In Collaboration with Deloitte. Technical report, Deloitte, 2020.

[23] J. Poon and T. Dryja. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. Technical report, 2016.

[24] M. Swan. *Blockchain : blueprint for a new economy*. O'Reilly Media, 1 edition, 2015.

[25] S. Underwood. Blockchain beyond bitcoin. *Communications of the ACM*, 59(11):15–17, oct 2016.

[26] H. Wang, Y. Cen, and X. Li. Blockchain Router. In *Proc. of the 6th Int. Conf. on Informatics, Environment, Energy and Applications - IEEA '17*, pages 94–97, New York, New York, USA, 2017. ACM Press.

[27] G. Wood. POLKADOT: VISION FOR A HETEROGENEOUS MULTI-CHAIN FRAMEWORK. Technical report, 2017.

[28] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander. Where Is Current Research on Blockchain Technology?-A Systematic Review. *PLOS ONE*, 11(10):163–189, oct 2016.

[29] A. Zamyatin, M. Al-Bassam, D. Zindros, E. Kokoris-Kogias, P. Moreno-Sanchez, A. Kiayias, and W. J. Knottenbelt. SoK: Communication Across Distributed Ledgers. Technical report, Cryptology ePrint Archive, 2019.

[30] A. Zamyatin, Z. Avarikioti, D. Perez, and W. J. Knottenbelt. TxChain: Efficient Cryptocurrency Light Clients via Contingent Transaction Aggregation. Technical report, Cryptology ePrint Archive, 2020.

[31] A. Zamyatin, D. Harz, J. Lind, P. Panayiotou, A. Gervais, and W. Knottenbelt. XCLAIM: Trustless, Interoperable, Cryptocurrency-Backed Assets. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 193–210. IEEE, may 2019.

Selected Papers by the Author

K. Košťál, P. Helebrandt, M. Belluš, M. Ries, and I. Kotuliak. Management and Monitoring of IoT Devices Using Blockchain. *Sensors*, vol. 19, no. 4, p. 856, Feb. 2019.

K. Košťál, T. Krupa, M. Gembec, I. Veres, M. Ries, and I. Kotuliak. On Transition between PoW and PoS. In *Proceedings of 2018 International Symposium ELMAR: 60th International symposium.*, pages 207–210, Zadar, Croatia, 2018. IEEE Press.

V. Valaštín, K. Košťál, R. Bencel, and I. Kotuliak. Blockchain Based Car-Sharing Platform. In *Proc. of ELMAR-2019: 61st Int. symposium.*, pages 5–8, Zadar, Croatia, 2019. IEEE Press.

K. Košťál, R. Bencel, M. Ries, and I. Kotuliak. Blockchain E-Voting Done Right: Privacy and Transparency with Public Blockchain. In *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)*, pages 592–592. Beijing, China, 2019. IEEE Press

Instructions to the authors

Publishing procedure

All contributions are web-published. A contribution is published without unnecessary delay right after it has been accepted. Contributions are published on the fly in the current issue. It is at the discretion of the Editor-in-chief to determine, when the current issue is closed and a subsequent new one is open. There will be at least two issues in a year but it is left up to the Editor-in-chief to adjust periodicity of the Bulletin to actual needs.

Extended abstracts of theses is the primary type of article in the Bulletin. Each extended abstract will by annotated by identifying the thesis supervisor, who must recommend it for publication and stands for the Editorial Board in a role similar to a reviewer. We offer publishing extended abstracts on the Bulletin's web before the thesis is defended. This preliminary publishing is a specific service to the academic community. As soon as we learn about successful defence, the extended abstract gains the status of accepted paper and will be included in the forthcoming issue. The accepted paper will be annotated with the date of successful defence and name of the insitution where the defence took place.

It is the policy of the Bulletin to offer a free access to all its articles on the web. Moreover, the publisher will seek opportunities to promote as wide as possible access and/or indexing of the articles. All the past issues remain accessible on the web as part of the web portal of the Bulletin. Closed issues will be made available also in a printable form, free for downloading and printing by anyone interested.

Policy on Originality

It is the policy of the Bulletin that Slovak University of Technology be the sole, original publisher of articles. Manuscripts that have been submitted simultaneously to other magazines, journals or to conferences, symposia, or workshops without the prior written consent of the Editor-in-Chief will be rejected outright and will not be reconsidered. Publication of expanded versions of papers that have been disseminated via proceedings or newsletters is permitted only if the Editor-in-Chief judges that there is significant additional benefit to be gained from journal publication. A conference chairperson can arrange with the Editor-in-Chief to publish selected papers from conferences, symposia, and workshops, after suitable reviewing. The papers must meet the editorial requirements for research articles. Acknowledgement of the originating conference will appear as a credit when the paper is published in the Bulletin.

Manuscript information for extended abstracts of doctoral dissertations

All contributions are submitted electronically. Send your manuscript as LaTeXsources and .pdf files by e-mail to `editor.acm@fiit.stuba.sk`. Paper's length should be 6-12 pages. Please, use LaTeXstyle, which is available to download at bulletin web-page `http://slovakia.acm.org/bulletin/`.

Some remarks to the provided style:

- **Headings and Abstract**
  The heading must contain the title, full name, and address of the author(s), thesis supervisor, abstract of about 100-200 words.
- **Categories and Subject Descriptors**
  Define category and subject descriptors according to ACM Computing Classification System (see `http://www.acm.org/about/class/1998/`).
- **Keywords**
  Please specify 5 to 10 keywords.
- **Equations**
  You may want to display math equations in three distinct styles: inline, numbered or non-numbered display (we recommend the numbered style). Please make sure that your equations are clearly formulated and described.
- **Figures and tables**
  Figures and tables cannot be split across pages, the best placement for them is typically the top or the bottom of the page nearest their initial cite. To ensure this proper "floating" placement of figures/table, use the environment figure/table to enclose the figure and its caption.
- **References**
  Please use BibTeX to automatically produce the bibliography. If possible use abbreviations like: Proceedings – Proc., International – Int., Conference – Conf., Journal – J.
- **Selected papers by the author**
  This section is used for thesis extended abstracts. Please write down all publications which are related to your thesis.

Published by Slovak University of Technology Press,
Vazovova 5, 812 43 Bratislava, IČO: 00397687
on behalf of the ACM Slovakia Chapter
ISSN 1338-1237 (printed edition)
ISSN 1338-6654 (online)
Registration number: MK SR EV 3929/09