РЕЗЮМЕТА НА НАУЧНИТЕ ТРУДОВЕ И УЧЕБНИ ПОСОБИЯ

на доц. д-р инж. Христо Георгиев Вълчанов
за участие в конкурс за заемане на академичната длъжност: ПРОФЕСОР
по професионално направление
5.3 „Комуникационна и компютърна техника”
. научна специалност „Компютърни системи, комплекси и мрежи“
учебна дисциплина „Администриране на локални и Интернет мрежи“
към катедра „Компютърни науки и технологии“
Факултет по изчислителна техника и автоматизация
обявен от Технически университет – Варна,
ДВ, брой 29 от 31.03.2023г.

Резюметата на научните трудове и учебни пособия са организирани в раздели както следва:

	Трудове за участие в конкурса за „Професор“	**брой**
В.4	**Публикации равностойни на монографичен труд на тема „Изследвания в областта на приложението на SDN и блокчейн технологиите за изграждане на интелигентни решения за облачни услуги ”**	**14**
Г	**Публикации извън групата на монографичния труд**	**56**
Г.7	Публикации в реферирани и индексирани в световноизвестни бази данни с научна информация	22
Г.8	Публикации в нереферирани списания с научно рецензиране	34

В.4 Публикации равностойни на монографичен труд на тема „Изследвания в областта на приложението на SDN и блокчейн технологиите за изграждане на интелигентни решения за облачни услуги”

В.4.1 Veneta Aleksieva, Hristo Valchanov and Anton Huliyan, Application of Smart Contracts based on Ethereum Blockchain for the Purpose of Insurance Services, 8-9.11.2019, Varna, BIA 2019, p. 69-72, ISBN 978-1-7281-4754-3, IEEE Catalognumber: CFP19U85-ART, DOI: 10.1109/BIA48344.2019.8967468

Този доклад представя експериментално внедряване на интелигентни договори за застрахователни услуги, базирани на Ethereum блокчейн. Реализиран е децентрализиран крипто-жетон, базиран на стандарта ERC20 за интелигентен договор. Създаден е уеб-базиран интерфейс за продажби на тези крипто-токени. Представени са резултатите от експерименталните тестове.

Класическият процес на заявяване на щета може да бъде подобрен чрез използване на интелигентни договори и блокчейн технология. Информацията за щетата може да бъде изпратена от застрахования или директно от сензори, монтирани в обекта на застраховка (интелигентен актив) до автоматизираното приложение за обработка на искове. За съответните застрахователни полици, предоставени от интелигентния договор, клиентът ще получи потвърждение в реално време. Искът се обработва автоматично чрез интелигентен договор въз основа на бизнес логика, като се използва информацията, предоставена от застрахователя.

Този подход автоматично използва допълнителни източници (статистика, отчети) за оценка на претенциите и за изчисляване на загубите. В зависимост от застрахователната полица, интелигентният договор може автоматично да изчисли личната отговорност. В определени ситуации интелигентният договор може да активира допълнителна оценка на иска. Ако искът е одобрен, плащането към застрахования се инициира чрез интелигентен договор.

Предимствата на новия подход, базиран на интелигентни договори за блокчейн технологията, могат да се видят в няколко аспекта. Подаването на искове е опростено и автоматизирано. Благодарение на директния обмен на информация за щети между застрахователите, този подход елиминира нуждата от брокери и намалява времето, необходимо за разглеждане на исковете. Вградената бизнес логика в интелигентния договор за блокчейн елиминира необходимостта от регулатор на загуби за преразглеждане на всяка претенция (освен в конкретни ситуации). Застрахователят има достъп до произхода на щетата, което му помага да идентифицира потенциални опити за измама. Процесът на плащане на щета е автоматизиран от интелигентния договор на блокчейна, без да е необходим посредник за заявяване на искове. Предложеното решение с интелигентни договори за застраховки се основава на стандарта ERC20. Той е внедрен експериментално на Ethereum блокчейн. Резултатите от експериментите показват, че предложеното решение е напълно работоспособно по отношение на управление на автоматичните плащания по одобрени претенции за загуба.

B.4.2 V. Aleksieva, H. Valchanov and A. Huliyan, "Implementation of Smart-Contract, Based on Hyperledger Fabric Blockchain", 2020 21st International Symposium on Electrical Apparatus & Technologies (SIELA), Bourgas, Bulgaria, 2020, pp. 1-4, doi: 10.1109/SIELA49118.2020.9167043.

Докладът представя решение за създаване на интелигентен договор, базиран на Permission блокчейн, конкретно- Hyperledger Fabric.
Предложеният смарт-договор е реализиран на компютър с AMD Ryzen 5 2600 с 6 ядра/12 нишки, 3.4GHz, 16GB DDR4 3200Mhz и SSD Nvme 500GB, скорост на четене/запис 3500/2700 MB/s. Операционната система е Linux Ubuntu 16.04 LTS 64bit. Освен това се използват Docker Engine версия 17.03 и Docker-Compose версия 1.8.
Топологията на Blockchain мрежата е следната: Има една компания (R1), която има един order възел (O1) и един peer възел (P1). Работи с две различни компании - R2 и R3. Всеки от тях има собствен консорциум с основна компания. Те се реализират в два независими канала - C1 и C2. Всеки консорциум има два peers - C1 има P3 и P1, C2 има P2 и P1. Тъй като peer P1 работи за основната компания R1, той участва в два канала. L1 е копие на Blockchain на C1, L2 е копие на Blockchain на C2.
Бизнес решението, базирано на блокчейн, се реализира чрез осигуряване на връзка между отделните организации за съхранение и обмен на информация, както и за нейната обработка. Данните са видими само между организациите, които имат права на достъп, за които между тях са създадени канали за комуникация. За да се поддържа коректността на данните по време на запис и съхранение, пиърите се конфигурират в рамките на организацията, за да поддържат работоспособността на мрежата.
Блокчейн мрежата използва Docker контейнер за внедряването на Hyperledger Fabric. Той използва инструмента Docker Compose за определяне и изпълнение на многоконтейнерни приложения на Docker.
След като блокчейн мрежата е конфигурирана и стартирана, бизнес логиката, която ще се изпълнява в нея, трябва да бъде внедрена. Интелигентните договори (codechains) се създават с езика за програмиране Go.
Тестовете са представени с Hyperledger Explorer за Fabric 1.4.x под Linux Ubuntu. Предложеното решение позволява бърза и сигурна миграция на интелигентни договори между независими канали. Всеки канал има собствена бизнес логика и е невидим за участниците в други канали.

B.4.3 V. Aleksieva, H. Valchanov and A. Huliyan, "Implementation of Smart Contracts based on Hyperledger Fabric Blockchain for the Purpose of Insurance Services", 2020 International Conference on Biomedical Innovations and Applications (BIA), Varna, Bulgaria, 2020, pp. 113-116, doi: 10.1109/BIA50171.2020.9244500.

Докладът представя внедряване на интелигентни договори за застрахователни услуги на собственост, базирани на Hyperledger Fabric Blockchain. Частната блокчейн като Hyperledger Fabric Blockchain е по-добро решение за застрахователния бизнес, защото работи върху доверени устройства (nodes), не се налага изискване за протокол за консенсус. Основните предимства са бързият достъп до информация, по-евтините транзакции и контролът на ниво поверителност. Поради тези факти този блокчейн е подходящ и полезен в много области на застрахователните услуги.
В представения случай на използване се създават два канала: един за консорциум 1 (Channel 1) на компания Org1 (застрахователна компания) и компания Org2 (брокер 1), и един за консорциум 2 (Channel 2) на компания Org1 и компания Org3 (брокер 2). Всеки канал има свой собствен блокчейн, както и интелигентни договори (codechain), които работят самостоятелно с него. Всеки консорциум има двама участници. Двата канала работят паралелно и не са видими за участниците извън разрешените от правилата на консорциума. Един peer може да съдържа копие от блокчейна и интелигентни договори на повече от един канал.
Предложеният умен договор (codechain) е реализиран на компютър с AMD Ryzen 5 2600 6 ядра/12 нишки, с Linux Ubuntu 16.04. Освен това се използват Docker Engine версия 17.03 и Docker-Compose версия 1.8. Името на създадения интелигентен договор е *mycc* и е инсталиран на Peer 0 на Org1 в Channel 1. Съгласно внедрената бизнес логика е възможно клиентът да смени своя брокер. Това означава, че неговата полица трябва да се премести от един канал в друг канал. Има два възможни сценария, след като се копира - той да остане видим в channel 1, а промените, направени след копирането в channel 2, няма да бъдат видими. Другият сценарий е, че той ще бъде изтрит, така че вече няма да се вижда за участниците в channel 1.
Тестването на работоспособността на случая на използване се извършва чрез изпращане на заявки до инсталираните codechains и проверка на правилното им изпълнение. Инструментът Hyperledger Explorer се използва за визуализиране на създадената мрежа за този експериментален случай на използване. За да se намери информация за човек, който е записан в Blockchain мрежата, се изпълнява скрипта на функцията *queryOwnerByName* от интелигентния договор.
Интелигентните договори предоставят възможност за създаване на полици, наблюдение на тяхното състояние и чрез бизнес логиката, която може да бъде описана в тях се автоматизира процеса на обработване на застрахователни искове. Чрез интелигентни договори е възможно да се създаде застрахователна полица, да се определи застрахователният риск, да се изпълнят плащания по застрахователни искове. Блокчейнът също оптимизира процеса на презастраховане, както и операциите на брокерите. В области, където е необходим мониторинг от страна на предлагането, това ново решение ще подобри застрахователния процес.

B.4.4 V. Aleksieva, H. Valchanov and A. Huliyan, "Smart Contracts based on Private and Public Blockchains for the Purpose of Insurance Services", 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2020, pp. 1-4, doi: 10.1109/ICAI50593.2020.9311371.

Този доклад представя експериментално внедряване на интелигентни договори, базирани на Hyperledger Fabric Blockchain за застрахователни услуги, в сравнение с друга реализация на интелигентни договори, базирани на Ethereum Blockchain. Случаят на използване е еднакъв във всяка реализация: застрахователната компания (Org1) работи с четири компании - Org2 (broker 1), Org3 (broker 2), Org4 (broker 3), Org5 (broker 4). Всеки консорциум има двама участници - застрахователна компания и една брокерска компания. Компанията Org1 има собствен пиър Peer0, който участва в четирите консорциума и има копие от четирите смарт договора. Peer0 има основна роля в застрахователния процес, тъй като той управлява отношенията между застрахователните и брокерските компании във всеки консорциум. Предложеното решение е разработено с Metamask, Truffle и Ganache под операционната система MacOS High Sierra. Ganache създава локален блокчейн на базата на Ethereum, който може директно да изпълнява команди, както и да извършва тестове. Използва се Metamask, тъй като няма нужда да се изтегля локално копие на Blockchain. Връзката към сайта прави връзка с Ethereum. Metamask се грижи за всички заявки от и към Blockchain мрежата. Metamask може да изпълнява функция на Ethereum портфейл и да поддържа изпращане и получаване на Ethers и ERC20 токени. Truffle се използва за прилагане на интелигентния договор. Това е интегрирана система за компилиране на записаните интелигентни договори, която ги качва в мрежата на Ethereum.
Същият случай на използване в Hyperledger Fabric се основава на четири канала. Предложеният интелигентен договор, базиран на Hyperledger Fabric, е реализиран на компютъра с процесор AMD Ryzen 5 2600 6 ядра/12 нишки и с операционна система Linux Ubuntu 16.04. Освен това се използват Docker Engine версия 17.03 и Docker-Compose версия 1.8. Основната разлика от решението, основано на публичен блокчейн, където съществува мрежата, е, че в частния блокчейн първата стъпка е да се създаде блокчейн мрежата.
Предложеното публично решение с интелигентни договори за застраховка се основава на стандарта ERC20. Той е внедрен експериментално на Ethereum блокчейн. Резултатите от експериментите показват, че предложеното решение е напълно работоспособно по отношение на управлението на автоматичните плащания по одобрени искове за загуба. В предложения смарт договор бизнес логиката е по-сложна и решението е по-скъпо от решението, основано на частен блокчейн, тъй като трябва да се плати за изчислителна мощност с „ETH" токени.
Предложеното частно решение с codechains върху Hyperledger Fabric е по-гъвкаво, по-сигурно, по-бързо и по-евтино от предишното публично решение.

B.4.5 Veneta Aleksieva, Hristo Valchanov, Smart Contracts based on Hyperledger Fabric Blockchain for the Purpose of Health and Life Insurance Services, CIEES’21, AIP Conference Proceedings 2570, 020002, ISBN 978-073544375-4, DOI 10.1063/5.0099626

Този документ представя решение, основано на интелигентни договори на блокчейн, при което застрахователят плаща директно на болницата за услугите, предоставяни в полза на застрахования, и само ако застрахователната сума е недостатъчна, пациентът плаща на болницата.
Недостатъците на класическия процес в България от гледна точка на застрахователя са:
• Лицето трябва да плати директно за лечението си, което ще бъде възстановено след седмици или месеци.
• Във време, когато здравето на човека е основен приоритет, той/тя трябва да предостави документи и да посети застраховател, за да възстанови направените от него разходи, понякога многократно.
За да се избегнат тези недостатъци, решението, предложено в този доклад, е с интелигентен договор за блокчейн. Стъпките на процеса са:
1. Болест/злополука на застрахования, за която трябва да се приложи лечение.
2. Лечението включва медицински прегледи, болнично лечение, амбулаторно лечение (лекарства с рецепта, наблюдение на състоянието на застраховани от личния лекар, контролни прегледи от специалисти).
3. В случай, че лицето е задължително осигурено и/или доброволно осигурено, за заплащане на лечението, се сключва интелигентен договор, който проверява дали лицето е осигурено (ако да - нарежда покриването на сумите от НЗОК, съгласно одобрен списък със суми, които НЗОК покрива, като за останалата част от сумите за лечение проверява наличните застрахователни суми за лицето и нарежда покриването на сумите от съответния застраховател, като вписва в полицата на застраховани изразходваната сума и само в в случай че сумата на лечението не може да бъде покрита от НЗОК и застрахователя, лицето заплаща допълнително с директно плащане.
Предложената реализация се основава на Hyperledger Fabric. За всяко застрахователно дружество се създава собствен канал (консорциум). Всеки канал има своя собствена блокчейн, както и интелигентни договори (codechains), които работят само с него. Всеки консорциум има двама пиъри - Peer0 от Org1 и друг пиър от застрахователната компания. Четирите канала работят паралелно един с друг и не са видими за участниците извън тези, позволени от правилата на консорциума. Peer0 има отделни копия на четирите блокчейна. Бизнес логиката на блокчейн мрежата се реализира на езика Go.
С предложеното решение застрахователят ще избегне директни плащания. Това ще намали документите, ще премахне необходимостта от експерт за застрахователя, което ще намали оперативните му разходи и риска от едно застрахователно събитие да използва две полици с припокриване, а не с допълване. Представени са експерименталните резултати, които доказват приложимостта на предложеното решение.

B.4.6 D. Todorov, H. Valchanov and V. Aleksieva, "Load Balancing model based on Machine Learning and Segment Routing in SDN", 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2020, pp. 1-4, doi: 10.1109/ICAI50593.2020.9311385.

Този доклад предлага модел, който има за цел да намали не само общото натоварване на SDN мрежата, но и да намали честотната лента и да подобри механизма за маршрутизиране в SDN мрежите. Той съчетава алгоритъм за маршрутизиране на сегменти и механизми за балансиране на натоварването, базирани на невронни мрежи. Основната цел на този модел е да изследва най-съвместимия модел на невронна мрежа за балансиране на натоварването на мрежата и да минимизира мрежовия трафик между контролера и мрежовите устройства.
В SDN има различни механизми за балансиране на натоварването, които използват два основни подхода - статично и динамично балансиране на натоварването. Разглеждат се недостатъците и проблемите на подходите и алгоритмите в подобни решения.
Моделът е разработен като крос-платформен SDN контролер, написан на езика за програмиране на C ++. Той внедрява протокола *OpenFlow* и следва специфични за операционната система системни повиквания за по-висока производителност.
Системата съдържа четири основни модула: *SDN controller module, Prediction module, Path compute module* and *Path encoding module*. Събраните мрежови параметри от системата се използват за изчисляване на оптималния път въз основа на алгоритми на невронни мрежи. Параметрите се свеждат до единичен коефициент, който след това се използва за обучение и прогнозиране.
Използвайки Q-Learning алгоритъм, процесът е разделен на два потока. Първо, няма данни за прогнозиране. За да попълни тези данни, модулът започва да се учи и получава награда за всяко успешно свързване. След като моделът е обучен, модулът може да предвиди всяка промяна на потока. След като връзката се установи, контролерът изпраща пакет за проверка на състоянието, за да получи мрежовата информация на устройството.
Когато получава информацията, контролерът я съхранява в база данни *Network Global View*. След това контролерът и суича започват да обменят ехо пакети, за да проверят връзката между тях. Тези пакети се използват за проследяване на честотната лента за връзка с устройството.
Процесът на прогнозиране следи възможните промени в натоварването на мрежата. Ако открие такива, той изпраща сигнал към *Path Compute Module*, за да актуализира *Flow* таблиците с необходимите маршрути, за да балансира натоварването на мрежата. След като оптималните пътища бъдат изчислени, те се изпращат обратно и се инсталират на суичовете. Контролерът също така уведомява процеса на прогнозиране за изпуснати мрежови устройства, което автоматично ще задейства промените в таблицата *Flow*.
Предложеният архитектурен модел комбинира алгоритми на невронни мрежи със сегментно маршрутизиране за постигане на по-добра производителност и балансиране на натоварването на мрежата. Той подобрява QoS и предоставя възможност за предсказване на претоварване на мрежовите маршрути.

B.4.7 D. Todorov, H. Valchanov, V. Aleksieva, Shortest path routing algorithm with dynamic composite weights in SDN networks, ICAI’21,30.09.-02.10.2021, Bulgaria, pp. 193-197, doi: 10.1109/ICAI52893.2021.9639512 ISBN:978-1-6654-2661-9

В този доклад е предложен алгоритъм за най-кратък маршрут с динамични композитни тегла в SDN мрежи, използващ протокол OpenFlow. Алгоритъмът избира по-малко натоварения път въз основа на динамични тегла на node и edge, като наблюдава натоварванията на връзките между суичовете. За да открие топологията на мрежата, алгоритъмът използва LLDP за намиране на връзки между суичове и разчита на ARP съобщения за намиране на хостове, свързани с тях. По този начин той може да извършва маршрутизиране чрез призрачни суичове, което OpenFlow не поддържа.
Експерименталното проучване е направено под Windows OS, а симулаторът Mininet се използва за симулиране на топология на мрежата и трафика. Симулаторът Mininet работи на виртуална машина на същата хост машина, на която работи контролерът. Контролерът е разработен с помощта на език за програмиране C ++ и внедрява OpenFlow за управление на SDN.
Разгледаните алгоритми за маршрутизиране са:
• простият алгоритъм за маршрутизиране на авторите с откриване на връзки между хостове-източници и хостове-дестинации (от предишната ни работа, представена в доклади);
• алгоритъмът за маршрутизиране на Dijkstra, който маршрутизира трафика въз основа на минималния брой hops;
• Предложен алгоритъм за най-кратък маршрут с динамични композитни тегла.
Експерименталните резултати показват, че алгоритъмът се представя по-добре от другите 2 алгоритъма. По време на фазата на откриване на топология, той показва по-малък мрежов трафик дори с използването на LLDP. Това се дължи на огромния обмен на ARP съобщения между суичовете, използвани от алгоритъма за просто маршрутизиране за свързване на мрежови устройства.
Също така алгоритъмът постига успешно основната си цел - да се балансира натоварването на мрежовия трафик и да се осигури по-добър QoS. Той има малко увеличение на закъсненията по време на трансфер на пакети, но това се дължи на промени в потока на пакети по време на процеса на маршрутизиране, за да се предотвратят задръствания. По време на експериментите и трите алгоритма имат нулева drop rate и всички пакети се прехвърлят успешно. В допълнение, всички разгледани алгоритми за маршрутизиране успешно обработват мрежови цикли и имат ниско използване на паметта и процесорната мощност на контролера.

B.4.8 D. Todorov, H. Valchanov, V. Aleksieva, Simple routing algorithm with link discovery between source and destination hosts in SDN networks, ICAI’21,30.09.02.10.2021, Bulgaria pp. 188-191, doi: 10.1109/ICAI52893.2021.9639742, ISBN:978-1-6654-2661-9

В този доклад е представен прост алгоритъм за маршрутизиране с откриване на връзки между хостовете- източници и хостовете-дестинации в SDN мрежи, без да се взема предвид цената на връзката. Алгоритъмът намалява съобщенията, предавани между мрежовите устройства и контролера, както и изчисляването на пътя за потоците. За внедряването и тестването е разработен контролер OpenFlow, който извършва основните взаимодействия с мрежовите устройства и използва емулатор Mininet за извършване на изследвания.
Системата съдържа два основни модула: SDN контролерен модул и Path Compute модул. Модулът SDN Controller поддържа основните комуникационни функции между контролера и сиучовете Той съхранява информация за свързаните устройства и техните таблици на потоците в базата данни Global Network View. За да се установи връзка между контролера и суича, се обменят handshake пакети. След установяване на успешна комуникация, контролерът периодично изпраща ехо пакети за проследяване на достъпността на суича.
Модулът за маршрутизиране е отговорен да намери адреса на местоназначението във flow entry таблицата и да намери следващия hop за пакета на контролера. Не се взема предвид натоварването на мрежата за постигане на балансиран трафик. Модулът взема предвид първото обслужено ARP съобщение, въз основа на което взема решение къде да пренасочи пакета.
Експерименталното изследване се извършва под Windows OS. За симулиране на топология на мрежата и трафика се използва симулатор Mininet. Mininet работи на виртуална машина на хост машината. Контролерът с предложения алгоритъм за маршрутизиране работи на хоста. Контролерът е разработен с помощта на език за програмиране C ++ и внедрява OpenFlow за управление на SDN.
Експерименталните резултати показват, че алгоритъмът постига основната си цел да намали мрежовия трафик между контролера и мрежовите устройства по време на фазата на откриване. Алгоритъмът не използва Link Layer Discovery Protocol (LLDP) за намиране на връзки между мрежови устройства. По този начин елиминира допълнителния трафик и запазва мрежовата bandwidth. Алгоритъмът има нулева drop rate и всички пакети с прехвърлени успешно. Той също така показва ниски времена за прехвърляне на пакетите. Друго предимство е успешната обработка на мрежови цикли в топологията на мрежата и по-малкото използване на паметта и процесорната мощност от контролера.

B.4.9 Veneta Aleksieva, Hristo Valchanov, Monika Vangelova, Cloud Based System for Reservation of Medical Appointments, AIP, CIEES'21, AIP Conference Proceedings 2570, 020002, ISBN 978-073544375-4, DOI 10.1063/5.0099627

Този доклад представя cloud система за записване на часове за клинични прегледи и консултации от разстояние. Направено е сравнение между три различни решения. Експерименталните резултати показват, че предложеното облачно решение е най-добрият вариант по отношение на скоростта на реакция, мащабируемостта, най-лесно администриране и рентабилност.
Авторите предлагат уеб-базирана система *CollosalClinic_Online*. За реализацията се използват различни инструменти като C#, HTML, CSS, JavaScript, Bootstrap, jQuery, Google API, ASP.NET. Разработката е в интегрирана среда MS Visual Studio 2017, а управлението на релационната база данни е с MS SQL Server 2019. Уеб сървърът е Apache 2.4.46, а Internet Information Services 10.0 се използва за уеб приложението и управлението на сайта, контейнеризация и бърза облачна интеграция. Тества се на локален компютър.
Втората реализация е в разпределена среда с платформа VMware Workstation Pro v.12.5.1.
Третата реализация е в облака Azure. Достъпът до приложението се осъществява чрез Интернет с URL адрес, генериран от Microsoft Azure с домейн на Azure, *https://purple-forest-09d81c203.azurestaticapps.net*. Microsoft Azure позволява да се изгради табло за мониторинг на ресурсите и производителността на системата. Формира се основно табло за управление, в което се изграждат и коригират всички необходими графики за наблюдение в реално време.
Направено е сравнение между трите реализации. Резултатите показват, че cloud-базираното решение е най-бързото, най-ефективното, има отлична производителност и устойчивост на грешки. След сравняване на това решение с пет други съществуващи решения за записване на часове за медицински прегледи и според резултатите от измерването на времето за зареждане, изтеглянето на ресурси и броя на заявките към сървърите, където се хостват приложенията, cloud реализацията на предложената система има най-добри показатели за производителност.

B.4.10 D. Todorov, H. Valchanov, V. Aleksieva, Comparative Evaluation of Traffic Load Balancing and QoS in SDN Networks, AIP, CIEES'21, ISBN 978-073544375-4, DOI 10.1063/5.0099807

В този доклад се предлагат различни важни критерии за прилагане на сравнителна оценка на балансирането на трафика. В края е представен комплексен сравнителен анализ на алгоритми за статично и динамично маршрутизиране за балансиране на натоварването на трафика и подобряване на QoS в SDN. За статично маршрутизиране бяха сравнени три алгоритъма - алгоритъм Open Shortest Path First, shortest widest path и simple routing with link detection, предложено от авторите в други изследвания. За динамично маршрутизиране бяха сравнени три алгоритма - Extended Dijkstra's algorithm, Enhanced Interior Gateway Routing Protocol and dynamic routing with complex weights, също предложен от авторите в друго изследване.
Експерименталното проучване и резултатите са получени под Windows OS, а Mininet simulator се използва за симулиране на мрежова топология и мрежов трафик. На същата хост операционна система, на която работи контролерът, симулаторът Mininet се изпълнява като виртуална машина. За целите на експерименталното изследване, контролерът е разработен с помощта на език за програмиране C ++ и внедрява OpenFlow за управление на SDN мрежа. Контролерът поддържа основните функционалности за управление на SDN мрежа и са реализирани алгоритмите: OSPF, simple routing with link detection and dynamic routing with complex weights. Контролерът реализира всички актуални версии на комуникационния протокол OpenFlow и има модулен дизайн. Той съхранява изгледа на глобалната мрежа в оперативна памет и има възможност да инсталира правила за потока върху мрежовите ресурси, както и да обработва всеки пакет независимо, като използва входящи и изходящи пакети-съобщения. Контролерът има възможност да открие достъпността на суич с помощта на ехо съобщения, които се обменят на всеки 5 секунди. Той също така поддържа ARP съобщения за откриване на свързани хостове към мрежови ресурси и съхранява техните MAC адреси в оперативната памет, като съответства хоста към съответния суич, с който има физическа връзка.
Тестовете са направени с различни топологии за всеки алгоритъм за маршрутизиране.
Авторите предлагат система от критерии за сравнение и комплексна оценка на тези алгоритми за маршрутизиране. Ако критериите се спазват отделно, може да се види, че предложеният от авторите алгоритъм за статично маршрутизиране не дава добри резултати за „натоварване на мрежата" в сравнение с другите два алгоритъма, но има равни резултати с тях за поддържане на всички мрежови топологии. Също така, спазването на критериите за механизми за динамично маршрутизиране показва, че предложеният от авторите алгоритъм има равни резултати за „Packet drop rate", „Топологии" и „Поддръжка на QoS". Поради широкия диапазон от критерии в комплексната оценка, общата геометрична и аритметична комплексна оценка на двата предложени от авторите алгоритъма е по-добра от алгоритмите, с които се сравняват.

B.4.11 H.Valchanov, V.Aleksieva. Novel blockchain - based models for healthcare and life science solution, 2022 International Scientific Conference on Communications, Information, Electronic and Energy Systems, CIEES’22 (приета)

Здравната политика, свързана с ваксинирането срещу инфекциозни болести на всяка страна има за цел да ограничи епидемиите и да запази здравето и работоспособността на гражданите, да удължи и подобри живота им. Обхващането на цялото население с ваксинация е един от ключовите фактори за постигане на тези цели. Този доклад предлага няколко модела, базирани на блокчейн технологии, които гарантират ваксиниране на цялото население, игнорират възможността за дублиране и фалшифициране на информация, минимизират бумащината между институциите, участващи в процеса на ваксиниране, подобряват процеса на планиране на необходимите ваксини и др. Моделите са реализирани върху частен блочейн Hyperledger Fabric.
Първият модел е фокусиран върху пациента. За всеки гражданин в момента на неговото раждане (или имиграция) се създава отделен канал, номериран с персоналния идентификатор на пациента (PID), който действа като индивидуален имунизационен паспорт. Информацията за всяка ваксинация се записва в един отделен блок в канала (състои се от датата на ваксинацията, вида на ваксината, кода на флакона, кода на GP и съпътстващите реакции към ваксината.
Вторият модел се фокусира върху имунизационния календар за заболявания, за които се поставят задължителни ваксини. За всяка година се създава отделен канал. В него се съхранява информация за ваксини за хора, родени през тази година.
Третият модел се фокусира върху заболявания, срещу които се поставят задължителни ваксини. Създават се отделни канали за всяко от заболяванията, подлежащи на профилактика чрез задължителна имунизация. Смарт контрактът сравнява данните с информацията от централната система за гражданска регистрация (ЕСГРАОН) за ваксинираните лица и МЗ получава точна информация за необходимите ваксини. РЗИ ще получава информация за ваксинираните лица, които не са ваксинирани.
Предложените модели са изцяло в съответствие със здравната политика на ЕС за дигитализация и модернизация на здравните системи. Тези модели предлагат промяна в текущия документооборот между институциите чрез въвеждането на национален имунизационен регистър, базиран на блокчейн. По този начин те гарантират надеждността на извършените ваксинации както на лицето, така и проследяването на неваксинираните лица за конкретна задължителна ваксина. От друга страна, моделите осигуряват по-добро планиране на необходимите количества ваксини.

B.4.12 H.Valchanov, V.Aleksieva. Smart Contracts based on Hyperledger Fabric Blockchain for the Purpose of Higher Education Subsidizing, 2022 International Scientific Conference on Communications, Information, Electronic and Energy Systems, CIEES’22 (приета)

По време на следването си много студенти се отказват от една специалност, след което кандидатстват и записват друга. Държавата субсидира няколко семестъра от обучението им, след което отново субсидира обучението на студента в друга специалност. Редица студенти започват, но не завършват обучението си, но и това е субсидирано от държавата. Не е насочено изследване върху загубите на държавата от субсидиране на образованието в тези случаи. Този доклад предлага модел, при който всеки гражданин може да получи субсидия за обучението си, но само за до 10 семестъра. След това той трябва да заплати пълните разходи за обучението си. При този модел студентите са мотивирани да завършат обучението си в рамките на субсидираните семестри, а държавата не търпи загуби от рефинансиране на същия студент, чийто срок на обучение надвишава тези семестри. Моделът се основава на смарт контракт на платформата Hyperledger Fabric.
За всеки държавен университет има отделен канал. Съответният университет е може да чете и пише блокове . Министерството на образованието има достъп само за четене до този канал. Каналът записва във всеки нов блок информация за записан студент за съответния семестър. Смарт контрактът за съответен канал, изчислява размера на всяка субсидия според специалността на студента и според текущите коефициенти и базовата субсидия, определена за съответната година с наредби.
HyperLedger Fabric е реализиран чрез Docker платформа. Всеки от партньорите в мрежата Fabric работи в отделен контейнер. Контейнерите могат да работят заедно на една машина или всеки на отделен възел. Тъй като комуникацията се основава на набор от протоколи TCP/IP, това позволява прилагането на предложения модел на разпределени платформи. Бизнес логиката на предложения модел се изпълнява от смарт контракти, на базата на езика Go. Взаимодействието с тях се осъществява чрез извикване на техните методи. За съхраняване на информация се използват различни обекти.
Използването на специален контейнер за изпълнение на команди позволява ефективно тестване на разработения програмен код. Създадени са множество скриптове, съдържащи заявки, извикващи методи на интелигентни договори с различни данни. Това позволява да се автоматизира процеса на тестване чрез многократно изпълнение в режим на отстраняване на грешки.
Представени са експериментални резултати, които доказват приложимостта на предложеното решение. Целта на бъдещата работа е да се внедри моделът в разпределена среда и да се разработи API потребителски интерфейс към интелигентни договори.

B.4.13 H.Valchanov, V.Aleksieva. Blockchain and IoT integration for smart transportation, International Conference on Electronics, Engineering Physics and Earth Science (EEPES 2022), Journal of Physics: Conference Series, Volume 2339, pp.1-8, Online ISSN: 1742-6596, Print ISSN: 1742-6588, DOI: https://doi.org/10.1088/1742-6596/2339/1/012012

Безопасният транспорт на специални и опасни товари е от съществено значение за екологичната среда и човешкото здраве. Съвременните решения се базират на наблюдение на параметрите на техния транспорт със сензори в реално време, което позволява бърза реакция при неочаквани събития. Въпреки наличната информация от датчиците, процесът по доказване на застрахователно събитие и изплащане на обезщетение е както при останалите застраховки. Този документ предлага базиран на блокчейн и IoT модел, който записва хронологично данните от сензорите, разположени в превозното средство със стоки, и интелигентния договор, който изпраща навременни сигнали до заинтересованите страни (включително застрахователя), когато параметрите на сензорите надвишават зададените прагове. Представена е експериментална реализация върху HyperLedger Fabric, която доказва приложимостта на предложения модел.
Информацията за щетата може да бъде изпратена директно от сензорите, монтирани в застрахования обект (контейнер, превозно средство) към автоматизирано приложение за обработка на претенция. За съответните застрахователни полици, предоставени от интелигентния договор, клиентът ще получи обратна връзка в реално време. Искът се обработва автоматично от интелигентен договор, базиран на зададена бизнес логика, като се използва информация, предоставена от застрахователя. Интелигентният договор автоматично използва допълнителни източници (статистика, отчети), за да оцени иска и да изчисли щетите. В зависимост от застрахователната полица, интелигентният договор може автоматично да изчислява личната отговорност. Ако искът бъде одобрен, плащането към застрахования се инициира чрез интелигентния договор.
Тестването на предложения модел и неговото внедряване се извършва чрез изпращане на заявки към интелигентен договор. За визуализиране на мрежата и извличане на статистическа информация се използва уеб базиран инструмент – Hyperledger Explorer.
Предложеният модел предлага хронологично и прозрачно проследяване на данните от датчиците, разположени на контейнера/автомобила, като се изпращат навременни сигнали до заинтересованите страни, когато параметрите на датчиците превишават праговете. Предимствата са в намаляване на документацията и оперативните разходи на застрахователя в случай на застрахователно събитие, елиминиране възможността за измама, подобряване удовлетвореността на клиентите при разглеждане на искове. Получените резултати от експериментите доказват приложимостта на предложения модел.

B.4.14. H.Valchanov, V.Aleksieva. Novel Model for Hospitalization Tracking based on Smart Contracts and IoT, ICAI’22, pp.14-17, E ISBN:978-1-6654-7625-6, DOI: 10.1109/ICAI55857.2022.9959996

През последните години измамите и злоупотребите с фалшиви хоспитализации в здравеопазването се превърнаха в сериозен проблем, изискващ наблюдение на заетостта на болничните легла. Контролът ще повиши социалната, здравната и икономическата ефективност на разходите за здравеопазване, което от своя страна ще подобри качеството на здравните услуги. Докладът предлага нов модел, базиран на блокчейн. В този модел данните от сензори, разположени в болничните легла и фитнес тракер за всеки пациент се записват хронологично. В същото време, има както наблюдение на местоположението на тракера, така и наблюдение на жизнените показатели на пациента. Интелигентният договор изпраща своевременни сигнали до заинтересованите страни, когато параметрите на сензорите надвишават зададените норми.
Предложеният модел е част от решение за интелигентни болници, където пациентите са оборудвани със здравни устройства, които следят жизнените им показатели и ги споделят с други оторизирани потребители в блокчейн мрежата. Този модел може да проследява събития в реално време, без да може да бъде манипулиран.
Бизнес логиката на смарт контракта на НЗОК следи хоспитализациите в реално време и на тази база автоматично изчислява сумите за възстановяване към лечебното заведение. Така се осъществява постоянен контрол върху заявените суми за лечение от страна на болниците, елиминира се възможността за измами и се намаляват разходите на НЗОК за заявени, но реално неосъществени хоспитализации. По този начин средствата от НЗОК се изразходват прозрачно и само за реално извършени услуги.
Предложеният модел е реализиран върху частен блокчейн – HyperLedger Fabric и използване на Docker контейнери. Всяка болница е отделна организация с партньори чрез които се осъществява достъп до съответния блокчейн. IoT компонентите (сензорите) записват информацията за показателите на пациента в съответния канал – само те имат право да записват в каналите. Данните от каналите се четат от партньорите на съответните болници, както и от тези на застрахователните компании. НЗОК е партньор на всички канали и може само да чете информация от тях.
Проведени са редица експерименти и са представени получените резултати. Те ясно доказват приложимостта на предложения модел и неговите предимства пред традиционното решение за хоспитализации.
Предимствата на предложеният модел могат да се изразят в следните насоки: подобряване качеството на проследяване на здравословното състояние на хоспитализирани пациенти, намаляване обработката на документи и оперативните разходи, елиминиране възможността за измами и предлагане на по-добър контрол върху хоспитализациите.

Г. Публикации извън групата на монографичния труд

Г.7. Публикации в реферирани и индексирани в световноизвестни бази данни с научна информация

Г.7.1 Veneta Aleksieva, Hristo Valchanov and Diyan Dinev, Comparison Study of Prototypes based on LiFi Technology, 8-9.11.2019, Varna, BIA2019, p.73-76, ISBN 978-1-7281-4754-3, IEEE Catalog number: CFP19U85-ART, DOI: 10.1109/BIA48344.2019.8967478

В този доклад е предложен LiFi прототип за комуникация на данни и сравнително проучване на предложения от авторите LiFi прототип с други подобни LiFi прототипи (P.Goswamis и LiFiNano). Ограниченията на предложения прототип са:

• Максималното разстояние на предаване е когато LED крушката излъчва под ъгъл 90^{o} с хоризонтална равнина. Този ъгъл гарантира максималното разстояние на предаване от 80 см. Този резултат се постига в напълно тъмна стая.

• Колкото повече се намалява ъгъла на излъчване на LED крушката, толкова повече намалява разстоянието за успешно предаване. Когато ъгълът е по -малък от 40^{o}, получаването на данни е неуспешно.

Изследването на предложения прототип се фокусира върху въздействието на някои фактори на околната среда (като осветеност на слънчевата светлина, стъклена преграда и солена вода). Експериментите с прототипа са направени, за да се определят неговите ограничения за максимално разстояние на предаване при различни условия на околната среда.

Целта на настоящото изследване е да се съберат данни за предаване през различна среда. Използва се LiFi прототип, разработен в предишно изследване, но в софтуера са направени някои подобрения, като например подобряване на скоростта на предаване и корекция на грешки.

Директната слънчева светлина (в този експеримент - 7520 лукса) води до 100% загуба на предадената информация към приемника, дори ако е на 1 см от предавателя. Намалява се осветеността на слънчевата светлина, като се отдалечава прототипа от прозореца, тогава разстоянието Dmax се увеличава. При достигане на 2,5 м (200 лукса), Dmax е 60 см. Стойността на Dmax от 80 см се достига на разстояние 4,0 м от прозореца (където въздействието на слънчевата светлина е 0 лукса). Това е същото като Dmax, което се достига в напълно тъмна стая.

Ако дебелината на стъклена преграда е само 2 мм, разстоянието е същото като разстоянието без преграда. Но ако дебелината на стъклената преграда расте, Dmax намалява. При стъклена преграда от 12 см Dmax е само 40 см - половината от максималното разстояние.

Бяха проведени експерименти с прясна и солена вода (10% и 20% солен разтвор). Въздухът е изключен, тъй като модулите на предавателя и приемника са залепени за стъклото на аквариума. На разстояние 55 см само при 0% соленост има комуникация, но на разстояние 26 см има комуникация и при 20% солен разтвор.

Избрани са основни показатели за ефективност за оценка и сравнение на прототипа, предмет на гореспоменатите експерименти, и други прототипи на LiFi. Въз основа на тях се прави сравнението. Прави се комплексна оценка - средна аритметична и средна геометрична. По отношение на резултатите от комплексните оценки може да се заключи, че авторовият LiFi прототип е най - добрият вариант за целите на настоящото изследване.

Г.7.2 Haka, V. Aleksieva and H. Valchanov, "Comparative Analysis of Traffic Prioritisation Algorithms by LTE Base Station Scheduler," 2020 21st International Symposium on Electrical Apparatus & Technologies (SIELA), Bourgas, Bulgaria, 2020, pp. 1-4, doi: 10.1109/SIELA49118.2020.9167040.

Този доклад представя изследване на въздействието на предложения от авторите алгоритъм за приоритизиране на трафика в LTE мрежа, Round Robin (RR), Maximum Rate (MAX-Rate), Proportional Fair (PF), Exponential/Proportional Fair (EXP-PF) и тези, предложени от Myo и Akyıldız за QoS в 4G LTE безжична мобилна мрежа.
Сравнението се основава на резултатите от пропускателната способност, забавянето, коефициента на предаване на пакети (PDR) и коефициента на загуба на пакети (PLR). За да се проучи въздействието на алгоритмите за приоритизиране на трафика върху QoS, се използва продуктът за симулация на LTE, предложен и допълнително разработен от авторите.
Експериментални проучвания се провеждат за статични и мобилни UE за една LTE клетка, за която предавателната мощност е 40W (46.02dBm), 20 MHz честотна лента, мощността на шума е -160.99dBm, 100 налични PRB, 6 секторни клетки и радиус 770m. Брой потребители са съответно 20, 50, 70 и 100. Разстоянието на статичните UE до използвания eNodeB (m) е съответно 10, 90, 170, 250, 330, 410, 490, 570, 650 и 730 (55 м за всички мобилни UE). Изискваният вид услуга е GBR, задължителните RB от всяко UE са 5555 и плащат цена за гарантирана услуга със стойност 5. Скоростите на движение за мобилни UE (km/h) са съответно 10, 20, 30, 40, 50, 60, 70, 80, 90 и 100.
Представените резултати показват, че с по-малък брой абонати, предложеният алгоритъм осигурява по -високи стойности за изследваните параметри за статични абонати, разположени в обхват до 250 метра от eNodeB и осигурява по-високи стойности за изследваните параметри за мобилни абонати, движещи се с не повече от 80 км/ч. С увеличаването на броя на абонатите обслужването става равномерно, но за абонатите с най -висок приоритет се осигуряват по-добри стойности, докато за другите алгоритми резултатите са почти еднакви.
Предимството на предложения алгоритъм пред другите е, че той обслужва заявки с висок приоритет от абонати на по -близко разстояние до eNodeB и заявки от мобилни абонати. Обслужването на заявки от абонати, разположени по-близо до eNodeB е с по-добро QoS, тъй като качеството на канала на тези абонати е по-добро грешките при предаване са по -малко, което води до по -бързо обслужване. Приоритетната услуга за заявки от мобилни потребители подобрява QoS за тях, тъй като това намалява загубите на пакети при предаване. Разпределянето на повече ресурси към потребителите с по -висок приоритет ще ускори обслужването на техните изисвкания и освободените от тях ресурси ще се използват за обслужване на UE с нисък приоритет.

Г.7.3 Haka, V. Aleksieva and H. Valchanov, "Comparative Analysis of Traffic Prioritisation Algorithms in 6LOWPAN Networks," 2020 21st International Symposium on Electrical Apparatus & Technologies (SIELA), Bourgas, Bulgaria, 2020, pp. 1-4, doi: 10.1109/SIELA49118.2020.9167116.

Този доклад представя цялостен сравнителен анализ между предложения от авторите алгоритъм за приоритизиране на трафика на 6LoWPAN сензорна мрежа и пет стандартни алгоритъма на сензорната мрежа. Има два основни класа алгоритми за приоритизиране на трафика за сензорни мрежи: Knowledge Free и Knowledge Based. В алгоритъма на авторите, според първоначалното приоритизиране, планирани заявки с най-висок приоритет съдържат Emergency Dispatch Header. Той идентифицира пакета като спешен. В случай на множество пакети със Emergent Dispatch Header или обикновени пакети, заявките от мобилни устройства се обслужват с по-висок приоритет. Когато са налични много подвижни устройства, техните заявки са приоритизират, като се използва скоростта им на движение. С по -висок приоритет се обслужват заявки от по -бързо движещи се устройства. При наличието на множество мобилни устройства, движещи се с еднаква скорост, следващият критерий, по който заявките се приоритизират, е разстоянието на сензора до координатора. За тази цел е използван принципът на Least Weighted Farthest Number Distance Product First mechanism. По-висок приоритет имат пакетите, изпратени от най-близките до координатора сензори. Когато на еднакво разстояние до координатора има много сензори, заявките се приоритизират, като се използва типа на сензора. С най-висок приоритет са приложенията за здравни грижи, след това са за сигурност и наблюдение, мониторинг на околната среда, проследяване на животни, проследяване на превозни средства, земеделие и интелигентни сгради.
Авторите са създали симулатор, който се използва за изследване влиянието на предложения алгоритъм и стандартни алгоритми за приоритизиране на трафика върху QoS в една и съща сензорна мрежа.
Представен е подробен сравнителен анализ на предложения от авторите алгоритъм за приоритизиране на трафика за 6LoWPAN и пет други. За комплексно сравнение на алгоритми за приоритизиране на трафика в 6LoWPAN е предложена система от критерии. Сравнение на алгоритмите за приоритизиране на трафика се прави по закъснение, пропускателна способност, Packet Delivery Ratio и Packet Loss Ratio. Това сравнение е направено за конкретен тип трафик, за определени крайни възли.
Предложеният от авторите алгоритъм за приоритизиране на трафика в 6LoWPAN е по-добър от другите изследвани, според средните аритметични и средните геометрични комплексни оценки.

Г.7.4 Haka, V. Aleksieva and H. Valchanov, "Software Tool for Evaluation of Traffic Prioritisation Algorithms in 6LOWPAN Network," 2020 21st International Symposium on Electrical Apparatus & Technologies (SIELA), Bourgas, Bulgaria, 2020, pp. 1-4, doi: 10.1109/SIELA49118.2020.9167147.

Този доклад представя подобрения в симулационния продукт за 6LoWPAN мрежи, предложен от авторите, който дава възможност за изследване на качеството на услугите. Разгледано е влиянието на различни алгоритми за приоритизиране на трафика върху пропускателната способност, забавянето, packet delivery ratio и packet loss ratio. Включените алгоритми в софтуерния инструмент са: предложения от авторите алгоритъм и класическите алгоритми за приоритизиране: First Come First Served (FCFS), Least Number of Sensors First (LNSF), Least Number of Hops First (LNHF), Least Number Distance Product First (LNDPF), Least Weighted Farthest Number Distance Product First (LWFNDPF). Софтуерният инструмент предоставя интерфейс за оценка на предложения и класическите алгоритми в сензорните мрежи.
В предложения симулатор броят на сензорните възли, работещи в даден регион, може да варира до 100, в зависимост от размера на зоната, която трябва да бъде покрита. Устройствата в тази област могат да бъдат напълно функционални или с намалена функционалност. Напълно функционалните устройства могат да работят както като координатори, така и като крайни възли, докато тези с намалена функционалност работят само като крайни устройства.
6LoWPAN сензорна мрежа е симулирана с едно напълно функционално устройство, което обслужва заявките на крайните устройства. Целта на изследването е да се определи ефективността на алгоритмите, вградени в симулатора за приоритизиране на трафика и в кои ситуации, за кои възли те подобряват QoS.
Резултатите от предложения алгоритъм за приоритизиране показват, че стойностите за изследваните параметри са по-добри за статичните устройства, които са по-близо до координатора. Приоритизирането на заявките от възли, които са по-близо до координатора в сензорните мрежи, е важно, тъй като те са мрежи от множество устройства, които предават данни постоянно. Това причинява смущения в комуникационната среда и грешки, което инициира повторното изпращане на пакети. В резултат на това натоварването и забавянето на комуникацията се увеличават и влошават QoS. С по-малко устройства, заявките с най -висок приоритет се обслужват с повече ресурси - от възлите, разположени до 6 метра от координатора. Това ускорява обслужването за тези възли, като същевременно освобождава ресурси за използване за устройства с нисък приоритет и компенсира закъсненията. В случай на недостатъчни ресурси, заявките на устройства с най-нисък приоритет се отлагат за обслужване в следващия интервал от време.
Резултатите за мобилните възли съгласно предложения алгоритъм за приоритизиране показват, че стойностите за изследваните параметри са по-добри за възлите, движещи се със скорости над 3 m/s.

Г.7.5 D. Dinev, V. Aleksieva and H. Valchanov, "Study of Li-Fi Indoor Network Reliability", 2020 21st International Symposium on Electrical Apparatus & Technologies (SIELA), Bourgas, Bulgaria, 2020, pp. 1-4, doi: 10.1109/SIELA49118.2020.9167053.

В този доклад е предложено внедряване на тестова Li-Fi мрежа на закрито. Реализираната мрежа се състои от три LiFi точки за достъп в стаята за предаване на информация на разстояние 2,5 м от пода. Всяко от устройствата LiFi е на разстояние 75 см от съседното устройство. Максимален ъгъл на осветяване, при който устройствата предават информация е 45^{o}.
Целта е да се реализира хендовер от потребителско оборудване (UE) между LiFi точки за достъп на физически изградената мрежа, като се вземат предвид правилно и неправилно получените данни по време на този процес. Реализацията на хендовер в Li-Fi мрежи е много важна за повишаване на надеждността на мрежата и предаване на всички данни преди напускане на мрежата.
Бяха проведени следните експерименти:
- да се определи работоспособността на мрежата;
- да се извърши хендовер на потребители от една точка за достъп до друга съседна;
- докладване на правилно и неправилно получени данни;
По време на експериментите светлината в стаята беше средно 16,6 lx.
За първата група експерименти бяха използвани следните параметри:
- брой изпратени знаци - 10 000;
- скорост на движение на потребителското оборудване - 1m/s, 2m/s и 3m/s
- разстояние между предавателя и приемника (L) - 0,5 м, 0,8 м, 1 м, 1,2 м и 1,5 м;
От резултатите, получени чрез експериментите, може да се види, че при нормална скорост от 1 m/s всички данни, изпратени от предавателя, са били успешно получени без загуби или погрешно получени пакети при преминаване от една точка на достъп до друга. Това е така за всяко от измерените разстояния между предавателя и приемника. С увеличаване на скоростта се наблюдава увеличаване на процента на неправилно получени или неполучени данни.
За втората група експерименти бяха използвани следните параметри:
- брой изпратени знаци - 100 000;
- скорост на движение на потребителското оборудване - 1m/s, 2m/s и 3m/s
- разстояние между предавателя и приемника (L) - 0,5 м, 0,8 м, 1 м, 1,2 м и 1,5 м;
От резултатите, получени чрез експериментите, може да се види, че при нормална скорост от 1 m/s почти всички данни, изпратени от предавателя, се получават успешно. Загубите се дължат на факта, че за тази скорост устройството вече е напуснало мрежовия обхват и не е получило останалите пакети. С увеличаване на скоростта се забелязват все повече неправилно получени или неполучени данни.
Резултатите показват, че с увеличаване на скоростта на движение на потребителското устройство и разстоянието между приемника и предавателя процентът на погрешно приетите символи се увеличава. Скоростта, с която няма грешки по време на предаването в тази тестова Li-Fi мрежа, е 1m/s.

Г.7.6 Haka, V. Aleksieva and H. Valchanov, "Enhanced Simulation Framework for Visualisation of IEEE 802.15.4 Frame Structure on Beacon Enabled Mode of ZigBee Sensor Network," 2020 International Conference on Biomedical Innovations and Applications (BIA), Varna, Bulgaria, 2020, pp. 109-112, doi: 10.1109/BIA50171.2020.9244507.

Този доклад представя подобрения в симулационния продукт за ZigBee за IoT, предложен от авторите в предишно изследване. Основните подобрения на симулационния софтуер са: възможност за изчисляване на стойностите за Received Signal Strength (RSS) и Received Signal Strength Indicator (RSSI); визуализиране на съдържанието на кадъра IEEE802.15.4 в beacon-enabled режим; изучаване на класически алгоритми за приоритизиране на трафика в сензорни мрежи; проучване на параметри, влияещи на QoS, като Packet Delivery Ratio (PDR), Packet Loss Ratio (PLR), забавяне и пропускателна способност.
Като подобрение на симулационния продукт за мрежата ZigBee е внедрен един Knowledge Free и един Knowledge Based алгоритъм за приоритизиране на мрежовия трафик. Knowledge Free алгоритмите обработват заявките по реда на тяхното пристигане. Такъв алгоритъм за приоритизиране на трафика е First Come First Served (FCFS). Knowledge Based алгоритмите използват или информация за приложението, или информация за мрежата или и двете, за да дадат приоритет на трафика. Реализираният алгоритъм Least Number of Hops First (LNHF) се основава на познаване на мрежовата информация. Според този алгоритъм заявките от устройствата по-близо до координатора се обслужват с висок приоритет. Изграждането на ZigBee мрежа се осъществява с помощта на графичен потребителски интерфейс, чрез който се създават координаторите и към тях се добавят крайни сензорни възли. Параметри като: брой на свързаните крайни възли, честотна лента на канала, област, честота, ред на beacons и ред на суперкадри се задават за всеки PAN координатор. За да се уточни и свърже създадената симулация с ограниченията за определен регион в света, е добавена опция за избор на определен канал и визуализиране на работната честота.
Когато координаторът и крайните възли са правилно добавени със съответната конфигурация, трафикът, генериран от крайните възли в мрежата, се приоритизира. При приоритизиране на трафика според избрания алгоритъм се попълва съдържанието на пет IEEE 802.15.4 кадъра.
Резултатите за статични възли от проведените тестове показват, че алгоритъмът LNHF подобрява QoS за крайните възли, на разстояние до 7 м от обслужващото устройство. Това ще ускори работата, тъй като смущенията в тези възли са по -малко, тъй като сигналът от координатора е по -добър, съответно препредаването на пакети ще бъде по -малко.
Резултатите от тестовете с мобилни възли за разглежданите алгоритми за приоритизиране са подобни. За устройствата, движещи се със средна скорост, разпределените ресурси са малко и разглежданите стойности се влошават. Това може да влоши QoS за тези устройства, тъй като обработката на техните заявки ще бъде забавена, а допълнително забавяне ще бъде причинено от инициирането на предаване, когато устройството е извън обхвата на текущия координатор.

Г.7.7 Haka, V. Aleksieva, H. Valchanov and D. Dinev, "Analysis of ZigBee Network Using Simulations and Experiments", 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2020, pp. 1-4, doi: 10.1109/ICAI50593.2020.9311328.

Този доклад сравнява резултатите за стойностите на Received Signal Strength Indicator (RSSI) от възлите на крайните сензори, получени чрез симулиране на мрежа ZigBee, използвайки подобренията на симулационния продукт чрез истинска сензорна мрежа ZigBee. Графичният потребителски интерфейс на симулатора позволява добавяне на координатори и крайни сензорни възли за изграждане на мрежа ZigBee. След като са добавени ZigBee PAN координатори, крайните сензорни възли могат да бъдат свързани към тях. Стойностите за RSS и RSSI се изчисляват веднага след задаване на разстоянието на сензора от координатора. Промените за всички въведени параметри се отразяват в таблиците с данни и могат да бъдат проверени от раздела „Nodes Table". Симулаторът изчислява автоматично стойностите за RSS и RSSI, въз основа на разстоянието между добавените крайни сензори и PAN координатора. Изчислените стойности, са представени с графики според разстоянието на възела до PAN или идентификатора на възела.
Тестовете за RSS и RSSI от симулатора бяха получени след изграждане на мрежа ZigBee от един координатор (маршрутизатор ZigBee) и 6 сензорни възела ZigBee, свързани в топология „звезда".
Физическото изграждане на мрежата ZigBee се осъществява с платка BeagleBone Black-BBB01-SC-505 с операционна система Bone-Debian-7.8, работеща като ZigBee Gateway, трансивер Texas Instruments (TI)-CC2531EMK и TI мулти-стандартни сензорни възли-CC2650STK. ZigBee Gateway е конфигуриран с помощта на TI Z-Stack Linux Gateway. Платката CC2531EMK е конфигурирана да работи като трансивер ZigBee и сензорните възли за работа в мрежата ZigBee, използват CC-DEVPACK-DEBUG на TI. Прехвърлянето на данни и получаването на RSSI стойности от крайните сензорни възли във вече изградената мрежа ZigBee може да бъде проследено, когато втори трансивер CC2531EMK е конфигуриран да работи като ZigBee sniffer. Резултатите за получените RSSI стойности от изградената мрежа ZigBee са противоречиви при тестовете за 2, 4 и 6 сензорни възли. Резултатите от 2 сензора показват, че с увеличаване на разстоянието от координатора получените RSSI стойности се влошават. Тази тенденция не се наблюдава при тестовете с 4 и 6 сензорни устройства. В тях, с увеличаване на разстоянието от координатора, получените RSSI стойности са идентични или по -добри за някои от възлите и по -лоши за други. Това се дължи на наличието на външни шумови влияния и смущения между сензорните възли, които могат да се увеличат с броя на устройствата в мрежата. Получените резултати показват, че за 2 крайни устройства в мрежата стойностите за RSSI, получени чрез симулатора, са почти идентични с тези за тестовете с реална мрежа. Резултатите с 4 и 6 крайни устройства, получени чрез симулатора, са близки до тези на реалната мрежа. Отклонението в RSSI стойностите на симулатора е около 10dB в сравнение с действителните резултати.

Г.7.8 D. Dinev, V. Aleksieva and H. Valchanov, "Simulation Framework For Studying Quality of Service Traffic Prioritization Algorithms in Li-Fi Network", 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2020, pp. 1-4, doi: 10.1109/ICAI50593.2020.9311358.

Този доклад представя симулационен софтуер за изучаване на алгоритмите за QoS приоритизация на трафик в Li-Fi мрежи. В него са внедрени алгоритъм, предложен от авторите от предишни изследвания, алгоритъм за приоритизиране на трафика на Wang и два класически алгоритма - First Come First Served (FCFS) и Least Number of Hops First (LNHF). Симулаторът изчислява packet delivery ratio (PDR), packet loss ratio (PLR), пропускателна способност и закъснение въз основа на разпределението на ресурсите чрез внедрените алгоритми.
Предложената стратегия за приоритизиране на трафика разпределя ресурсите в рамките на един времеви интервал, като отговаря на някои критерии. Пренареждането на потребителите, свързани към терминала според алгоритъма на авторите, е следното: потребителите, които са по -близо до него, имат по -висок приоритет и отиват по-високо в таблицата на потребителите. Ако разстоянието от терминала до някои от потребителите е равно, тогава алгоритъмът търси следващи критерии - клиентското устройство мобилно или статично е? Статичните устройства имат по-малък приоритет. Мобилните потребители имат по-висок приоритет според скоростта си. Колкото по-висока е скоростта, толкова по-висок приоритет има устройството. Видът на исканата услуга е последният критерий на алгоритъма. Всеки от тях принадлежи към определен клас, който има различен приоритет според QoS параметрите. Има четири вида класове:
• Клас 1 - съдържа услуги за хендовер между клетки, повиквания за възстановяване на връзки и гласови повиквания.
• Клас 2 - съдържа услуги на видео повиквания.
• Клас 3 - съдържа услуги за предаване, HDTV и гласови съобщения.
• Клас 4 - съдържа само услуги за фонов трафик.
Новата функционалност включва възможност за приоритизиране на свързани потребители чрез внедряване на нови алгоритми, изчисляване на техните QoS параметри за PDR, PLD, закъснение и пропускателна способност, сравняване на параметрите по всеки алгоритъм и показване на предавателната матрица за всеки алгоритъм. Предавателната матрица за всеки алгоритъм може да бъде показана след изчисляване и разпределение на ресурсите, поискани от свързаните устройства. Добавена е нова таблица с данни за съхраняване на параметъра QoS за всеки алгоритъм. За лесно сравняване и проучване на QoS параметрите за всеки алгоритъм може да се направи графична диаграма за всеки от тях.
Софтуерът реализира напълно работеща симулация на Li-Fi мрежа с терминални устройства и свързани с тях потребители с техните спецификации. Съгласно реализирания алгоритъм за приоритет на трафика и разпределение на ресурсите, софтуерът може да изчисли PDR, PLD, закъснение и пропускателна способност, които са важни за осигуряване на по-добро качество на услугата.
Според тези резултати може да се направи заключението, че алгоритъмът, който се предлага, има по-добри стойности на QoS от останалите, разгледани в тази статия.

Г.7.9 Aydan Haka, Veneta Aleksieva, Hristo Valchanov, 6LoWPAN Network Analysis Using Simulations and Experiments, IOP Conference Series: Materials Science and Engineering, CIEES'20, vol. 1032 (2020), pp.1-4, doi:10.1088/1757-899X/1032/1/012015, ISSN: 1757-8981, https://iopscience.iop.org/article/10.1088/1757-899X/1032/1/012015

Този доклад представя физическо реализиране на 6LoWPAN мрежа и изследване на показателите за пропускателна способност от край до край, което се сравнява с резултатите, получени чрез симулационния продукт 6LoWPAN, представен в предишни изследвания на авторите. Тестовете за пропускателна способност и закъснение от край до край от симулатора бяха получени след изграждане на мрежа 6LoWPAN от един координатор и 6 сензора 6LoWPAN, свързани в топология „звезда". Координаторът е конфигуриран да работи по канал 25. До 6 крайни сензорни възела 6LoWPAN могат да бъдат свързани към координатора. Всички крайни възли са статични, изпълняват един и същ тип приложение и се намират на еднакво разстояние от координатора (от 1м до 5м). Тестовете за отчитане на стойностите за пропускателна способност и закъснение от край до край са направени с 2, 4 и 6 сензорни възли, свързани към 6LoWPAN координатора. След като се добави информация за координатора и крайния възел, се извършва симулация за изпращане на определен брой пакети. След добавяне на пакетите към опашката за изпращане се изчисляват стойности за закъснение от край до край и пропускателна способност. Резултатите от проведените експериментални проучвания са голям брой, затова те са обобщени и представени в таблица. Тъй като симулаторът представя експериментите в идеални условия, на различни разстояния получените стойности са идентични. Разликата в проведените експерименти се получава от различния брой изпратени пакети.
Физическото изграждане на 6LoWPAN мрежата се осъществява с платка BeagleBone Black-BBB01-SC-505 с операционна система Bone-Debian-9.9, работеща като 6LoWPAN Gateway, TI трансивер-CC2531EMK и TI мулти-стандартни сензорни възли- CC2650STK. Прехвърлянето на данни и броя на битовете за получаване от крайните сензорни възли във вече изградената 6LoWPAN мрежа могат да бъдат проследени, когато втори трансивер CC2531EMK е конфигуриран да работи като 6LoWPAN снифер. Това е направено на Linux машина с помощта на програмата Sensniff за 6LoWPAN.
Експериментите са направени с 2, 4 и 6 сензора със симулатор и с реална мрежа при еднакви условия. Например, отклонението в симулираните резултати с 6 сензора от реалните за закъснение от край до край е средно 99% за 5, 10, 15 и 20 изпратени пакета, а за пропускателната способност е 98% за 5 пакета, 94% за 10 пакета, 86% за 15 пакета и 79% при 20 пакета. Резултатите от тестовете в реална мрежа са променливи, тъй като комуникацията между сензорите и координатора се влияе от фактори на околната среда като електромагнитни смущения, радиосмущения, грешки при предаване на пакети, други източници, работещи на същата честота, смущения между сензори и др.
Получените тенденции в резултатите от симулацията и реалната мрежа се доближават, което дава основание да се твърди, че симулационният продукт е подходящ за образователни цели.

Г.7.10 Aydan Haka, Veneta Aleksieva, Hristo Valchanov, Deployment and Analysis of Bluetooth Low Energy Network, IOP Conference Series: Materials Science and Engineering, CIEES’20, vol. 1032 (2020), pp.1-4, doi:10.1088/1757-899X/1032/1/012016, ISSN: 1757-8981, https://iopscience.iop.org/article/10.1088/1757-899X/1032/1/012016

Този доклад представя реализирането на физическа Bluetooth ниско енергийна (BLE) сензорна мрежа за IoT и изследване на RSSI стойностите, получени за крайните сензорни блокове в мрежата. Физическото изграждане на BLE мрежата е направено с платка RaspberryPi 4 Model B с операционна система Raspbian, работеща като BLE master устройство, с вграден BLE трансивър и многостандартни сензорни възли на Texas Instruments-CC2650STK. Топология “звезда” е реализирана чрез свързване на крайните сензорни възли и master-а за изследване на промяната в RSSI стойностите. Извършени са различни експерименти с 1, 2, 3, 4, 5 и 6 статични възли, където за всеки от тях възлите са разположени на разстояния от 1 м до 10 м от главното устройство. Извършено е изследване на промените в получените RSSI стойности за статични сензори, разположени на различни разстояния от главното устройство и за мобилни възли, движещи се с различни скорости.
За 1 възел резултатите показват, че с увеличаването на разстоянието на сензора от главното устройство, получените RSSI стойности се влошават. Стойността на 10 метра обаче е значително по-добра от предишните. Въпреки че само едно устройство предава в комуникационната среда, която не е натоварена, спадът на предишните стойности може да се дължи на външни източници на смущения. Тенденцията, че на по-близко разстояние до обслужващото устройство получените RSSI стойности са по -добри, се потвърждава от другите тестове с 3, 4, 5 и 6 сензора. Измерените стойности за RSSI намаляват все повече и повече, когато разстоянието от главното устройство и броя на крайните възли в мрежата се увеличават. Подобни експерименти са проведени и с мобилни възли. За втория възел се вижда, че стойностите за RSSI са значително по-ниски. Това се поражда от натоварването на комуникационната среда и възникналите смущения. Тенденцията, когато сензорите се движат с по-ниска скорост, получените RSSI стойности са по -добри, се потвърждава от другите тестове с 3, 4, 5 и 6 сензора. Експерименталните резултати за RSSI със статични сензорни възли показват, че с увеличаване на разстоянието между крайните възли и главното устройство, получените стойности се влошават със значителни промени. Експерименталните резултати за RSSI с мобилни сензорни възли показват, че с увеличаване на скоростта на крайните възли получените стойности се влошават, но промяната в резултатите е по-плавна.
Както за статични, така и за мобилни възли се запазва тенденцията за влошаване на RSSI стойностите с увеличаване на броя крайни сензорни възли в мрежата.

Г.7.11 A. Haka, V. Aleksieva and H. Valchanov, "Simulation Environment for Research of Algorithms for Traffic Prioritisation in ZigBee Network," 2021 17th Conference on Electrical Machines, Drives and Power Systems (ELMA), 2021, pp. 1-4, doi: 10.1109/ELMA52514.2021.9503088.

Този доклад представя симулационна среда, която позволява да се проучи влиянието на внедрените алгоритми за приоритизиране на трафика върху параметри, свързани с качеството на услугата (QoS) в мрежата ZigBee. Предложеният алгоритъм за приоритизиране на трафика за ZigBee е модификация на предходен предложен от авторите алгоритъм и е предназначен да работи в топология „звезда". Подобренията на продукта са способността да се изследва влиянието на различни алгоритми за приоритизиране на трафика върху параметри, тясно свързани с QoS, както и визуализация на изградената топология на мрежата. Симулационният продукт има модулна архитектура, а работата на отделните модули се контролира от ядрото му. Алгоритъмът проверява няколко критерия за приоритизиране на трафика в мрежа ZigBee. Първо се проверява за пакети, които са маркирани като спешни. При наличието на такива пакети те се обслужват с най-висок приоритет. Когато има няколко спешни пакета или те липсват, трафикът се приоритизира според това дали заявката е от мобилно или статично устройство. Заявките от мобилни устройства се обслужват с по-висок приоритет. Когато има пакети от повече от едно мобилно устройство, заявките се приоритизират според скоростта, с която се движат устройствата. Заявките с по-висок приоритет се обслужват от устройства, които се движат по-бързо. Друг критерий за приоритизиране при равни други условия е разстоянието на сензора от координатора. Заявките от сензори, по-близки до координатора, се обслужват с по-висок приоритет. Когато сензорите са на равно разстояние от координатора, техните заявки се приоритизират според стойността на cost. Заявки с по-висока стойност на cost се обслужват с по-висок приоритет. И накрая, заявките се приоритизират според приложението на сензора.
Извършените експерименти имат за цел да проучат влиянието на внедрените алгоритми за приоритизиране на трафика в мрежата ZigBee върху параметрите PDR, PLR, закъснение и пропускателна способност, които са важни за осигуряване на добро QoS. Представените експериментални резултати показват, че с увеличаване на броя възли услугата на предложения алгоритъм за приоритизиране на трафика в мрежата ZigBee става равномерна. За изследваните параметри обаче са предвидени по-високи стойности за по-близките до координатора възли. Това ще подобри QoS и ще ускори обслужването за тези устройства. Това ще освободи по-бързо заетия ресурс и ще позволи по-бързо да се обслужват заявките с най-нисък приоритет от най-отдалечените устройства.
Обратно, услугата на класическите алгоритми е значително равномерна, което натоварва цялата комуникация в мрежата и може да доведе до влошаване на QoS. В допълнение, предоставянето на повече ресурси от предложения алгоритъм за обслужване на заявки от възли с по -висок приоритет, за разлика от класическите, ще удължи живота им на батерията, тъй като консумацията на енергия е само в активни периоди, а броят им може да бъде сведен до минимум чрез ускоряване на обслужването.

Г.7.12 Aydan Haka, Diyan Dinev, Veneta Aleksieva, Hristo Valchanov, Comparative analysis of ZigBee, 6LoWPAN and BLE technologies for the Internet of Things, AIP, CIEES'21,25-27.11.2021, Rouse, Bulgaria, pp. 1-4, ISBN 978-073544375-4, DOI 10.1063/5.0099684

Този доклад представя реализацията на сензорна мрежа за IoT със сензори Texas Instruments CC2650STK, които могат да бъдат конфигурирани да работят с ZigBee, 6LoWPAN и BLE технологии. Извършени са експериментални проучвания на параметрите End_to_End Delay, Throughput и PLR за трите технологии. Въз основа на резултатите от експериментите е представено сравнение на същите между разглежданите технологии. Целта е в резултат на изследването да се формулират препоръки за най -подходящата технология за изграждане на сензорна мрежа за IoT с използваните сензорни възли.
Експерименталните изследвания за разглежданите технологии се реализират с различен брой едновременно свързани в мрежата статични сензорни възли (2, 4 и 6). Експериментите включват изчисляване на стойностите на параметрите End_to_End Delay, Throughput и PLR, които влияят на QoS, на разстояния между обслужващото устройство и сензорните възли от 1m, 2m, 3m, 4m и 5m, при изпращане на 5, 10, 15 и 20 пакета. За да се осигури сравнимост между получените резултати за изследваните технологии, във всички експерименти е използвана топология „звезда".
Според получените резултати стойностите за End_to_End Delay се увеличават с броя на крайните възли в разглежданите технологии, тъй като е необходимо повече време за обслужване на заявките на всички устройства. С увеличаването на броя на изпратените пакети се увеличават и стойностите, получени за End_to_End Delay, тъй като има повече заявки за обслужване в мрежата. При ZigBee в повечето експерименти минималната и максималната стойност за End_to_End Delay е по -добра от 6LoWPAN и BLE. Освен това в повечето експерименти получените стойности за ZigBee са постоянни и не се променят драстично с увеличаване на разстоянието между крайните възли и обслужващото устройство.
От получените резултати за PLR може да се види, че стойностите се увеличават правопропорционално на броя на възлите в мрежата за разглежданите технологии.
Следните препоръки могат да бъдат формулирани от експериментите и получените резултати:
• В приложения, където е важно стойностите за End_to_End Delay да са относително ниски и постоянни е по -добре сензорните възли CC2650STK да бъдат конфигурирани да работят с технологията ZigBee;
• В приложения, където се изискват постоянна throughput е по -добре сензорните възли CC2650STK да бъдат конфигурирани да работят с технологията ZigBee;
• Когато се изисква да се осигури по-висока производителност с по-голям брой възли в мрежата е по-добре сензорните възли CC2650STK да бъдат конфигурирани да работят с BLE технология;
• В приложения, където се изисква по-малка загуба на пакети, е по-добре сензорните възли CC2650STK да бъдат конфигурирани да работят с ZigBee или BLE технология, тъй като получените PLR стойности са изключително близки, но с по -ниски стойности, получени за ZigBee.

Г.7.13 A.Haka, Y.Yordanov, V.Aleksieva, H.Valchanov, Simulation Environment for Bluetooth Low Energy Network , ICAI’21,30.09.-02.10.2021, Bulgaria pp. 287-290, doi: 10.1109/ICAI52893.2021.9639521 ISBN:978-1-6654-2661-9

В днешно време с разширяването и усъвършенстването на комуникационните технологии предлаганите услуги се увеличават, като например технологии за широколентов интернет на нещата (IoT), а една от най-разпространените IoT технологии е Bluetooth Low Energy (BLE).
Този доклад представя симулационен продукт за изследване на комуникацията и съобщенията между Master и Slave в мрежата BLE, който може да се използва и в образованието. Може да се използва както за изучаване на основните функционалности на технологията, така и по време на обучение на място или онлайн.
Разработеният симулатор в катедра КНТ към Технически университет - Варна е с модулна архитектура. При зареждане на приложението се стартира основната функционалност на ядрото, която е добавяне на Master устройството и реализиране на неговата програмна логика за обработка на входящите пакети и съответния им тип PDU, както и изчакване за добавяне на Slave устройство и наблюдение на състоянието му (Standby, Advertising, Connected). Изпълнението на основната функционалност се контролира от класа "AppController".
За да се получи статистическа информация за времето, през което крайните устройства в мрежата са били в определено състояние, ядрото се обръща към модула за статистика, който се управлява от класа „DeviceStatisticsUtil“. Обработената информация чрез различните модули се визуализира чрез изградения графичен потребителски интерфейс (GUI).
След добавяне на Slave устройства, на всяко от тях може да се позволи да визуализира разстоянието до Master, да бъде премахнато от мрежата или да промени статуса му от Standby на Advertising.
Когато състоянието на Slave е Advertising, то започва да изпраща рекламни пакети по предназначените за това канали (37, 38 и 39). С това Slave изпраща броадкастни пакети в комуникационната среда, така че да може да бъде открито от Master в обхвата му и евентуално да се свърже с него. При преминаване към Advertising режим също започва проследяване на пакетите, предавани през комуникационната среда.
За да се сравни обменът на съобщения при установяване на връзка, изпращане на данни и прекратяване на връзката между Master и Slave устройства в BLE симулатора и реална среда, се конфигурира истинска BLE мрежа. За да се осигури сравнимост между резултатите от реалната и симулирана BLE мрежа, е реализирана експериментална топология от един Master и един Slave.
По време на симулацията някои от детайлите на комуникацията са пропуснати, за да се опрости разглежданият процес и да се улесни представянето му по време на обучението.
Симулаторът представя основните съобщения в изпълнението на процеса, което му позволява да се използва по време на обучението както на място, така и онлайн. Резултатите показват, че симулаторът може да се използва за представяне на акцентите в комуникацията между Master и Slave.

Г.7.14 D.Dinev, V.Aleksieva, H.Valchanov, K.Genov, Simulation Software For Finding Best Route in LoRaWan Network, ICAI’21,30.09.-02.10.2021, Bulgaria pp. 291-294, doi: 10.1109/ICAI52893.2021.9639718 ISBN:978-1-6654-2661-9

LoRaWan е с дълъг обхват, ниска мощност, нискоскоростна, безжична телекомуникационна система, популяризирана като инфраструктурно решение за IoT: крайните устройства използват LoRaWan чрез един безжичен gateway, свързан с Интернет. Той работи като прозрачен мост и предава съобщения между тези крайни устройства и централен мрежов сървър.
Тази статия представя симулационен софтуер за намиране на най-добрия маршрут в LoRaWan мрежи.
Depth First Search е алгоритъм за обхождане или търсене в структури от данни като "дърво" и "граф". За да се приложи алгоритъмът, се избира връх или възел на структурата, който се обозначава като корен и обхождането започва от него. Всички следващи върхове се посещават последователно в дълбочина до достигане на един, без наследници, след което се извършва търсене с връщане назад до достигане на нова крайна точка или след пълното обхождане - до корена. Оригиналната версия на алгоритъма е създадена през 19 век от Шарл Пиер Тремо за решаване на проблеми с лабиринта.
Симулаторът използва модифицирана версия на алгоритъма, който търси всички пътища само до едно крайно устройство, дефинирано като дестинация, за да намери всички възможни маршрути от определена частна локална мрежа към друга мрежа. В мулти-хопови мрежи може да има няколко маршрута с еднакви параметри. Първоначално с помощта на Hassle Free Route маршрутът се избира според параметъра за най-кратък път. Включена е уникална стойност, за да се даде приоритет на маршрутите. Уникалната стойност се съхранява в таблиците за маршрутизиране на устройствата като отрицателно, положително число или 0, където:
• отрицателно число - има голяма загуба на пакети по маршрута;
• положително число - маршрутът е добър;
• 0 - стойност по подразбиране; маршрутът не е оценен.
По-високата положителна стойност показва, че маршрутът е по-добър от останалите. Тя показва броя на успешните предавания по този маршрут. За всяко успешно предаване тази стойност се увеличава с 1, а за всяко неуспешно предаване се намалява с 1. Когато фрагмент съдържа emergent dispatch header, той се препраща към маршрута с най-високата уникална стойност. Тези фрагменти се приоритизират и изпращат по най-предпочитания път.
Средното време за изпращане на 51-байтов LoRaWan фрагмент е Ttrans = 6 ms, което включва времето за предаване и back-off таймера.
Симулаторът има 5 основни модула - GUI, Core, Creating topology, Topology modification, Finding best path between end devices.
За да се направи тест за намиране на „Най-добър маршрут“ между крайни устройства, е симулирана LoRaWan мрежа с 5 терминални и 4 крайни устройства.
Направени са тестове за намиране на най-добрия маршрут между крайни устройства и е доказано, че предложеният симулатор е напълно функционален и подходящ за изследвания на LoRaWan мрежи.

Г.7.15 D.Dinev, V.Aleksieva, H.Valchanov, Comparative Analisis of Li Fi Simulators for Purposes of the Education, ICAI’21,30.09.-02.10.2021, Bulgaria pp. 125-128, doi: 10.1109/ICAI52893.2021.9639691 ISBN:978-1-6654-2661-9

Li-Fi технологията за Internet of Things осигурява висока скорост, двупосочен и сигурен безжичен достъп. Това изисква проучване на качеството на услугата на тази технология. Това може да стане с помощта на симулационен софтуер, който ще намали разходите и времето за изграждане на такива мрежи. Тази статия представя информация за сравнителен анализ между предложения от авторите в предишни доклади симулатор и някои от най-известните симулатори (OptSim, Veins VLC, NS-2, NS-3, MATLAB) за изследване на качеството на услугата в Li-Fi мрежи. Предложена е система от критерии за извършване на сравнителен анализ на симулатори за Li-Fi мрежата. Този подход към изследванията на Li-Fi мрежата за IoT може да бъде въведен и в образованието.
Съществуващите симулатори на Li-Fi мрежи имат редица недостатъци, свързани с тяхната работа и функционалност. Тук те са представени.
Предложените критерии за сравнение на симулаторите LiFi са:
• Моделиране на различни алгоритми за приоритизиране на трафика в LiFi
• Моделиране на различни методи за разпределение на ресурси
• Симулация на мобилност
• Поддържане на GUI
• Визуално представяне на изследваната мрежа
• Анализ на получените резултати
• Лесен монтаж
• Мащабируемост
• Ръководство на потребителя/разработчика
• Програмен език
• Използване на паметта
• Лиценз за използване
Според представените резултати от изследването, поради широкия спектър от разглеждани критерии, най-подходящи за изследване на Li-Fi мрежи са MATLAB, OptSim и NS-3. Отделно проучване на критериите обаче показва, че предложеният от авторите симулатор осигурява по-добри стойности за показателите: „Лесна инсталация“, „Използвана памет“ и „Лиценз за използване“, които са изключително важни за образователни цели.
Критериите за сравнение не определят критериите "Визуализация на матрицата на предаване", т.е. за разлика от симулатора, предложен от авторите, никой от другите симулатори не предоставя тази възможност. Той осигурява сравними, с други симулатори, резултати спрямо много други критерии. Това доказва, че авторският симулатор е много подходящ за обучение и образователни цели.

Г.7.16 A.Haka, V.Aleksieva, H.Valchanov, ZigBee Simulation Framework for Studying the Formation of a Hierarchical Tree Topology , ICAI’21,30.09.-02.10.2021, Bulgaria pp. 257-260, doi: 10.1109/ICAI52893.2021.9639563 ISBN:978-1-6654-2661-9

ZigBee е една от модерните технологии за управление на IoT сензорни мрежи, тъй като осигурява високо качество на обслужване (QoS) и ниска консумация на енергия. Едно възможно решение за постигане на по-добър QoS в тези мрежи е да се използва ефективен алгоритъм за маршрутизиране на трафика. Тази статия представя подобрен симулатор, в който е реализиран алгоритъм за формиране на йерархична топология на ZigBee, базиран на приоритети, позволяващ йерархично маршрутизиране. Симулаторът предоставя възможност за анализ на резултатите от алгоритъма чрез визуална интерпретация на мрежовата топология.
ZigBee използва смесен механизъм за маршрутизиране, комбиниращ hierarchical tree routing protocol (HRP) и ZigBee ad hoc on-demand distance vector (Z-AODV). HRP е активен метод за маршрутизиране, чиято информация за маршрута се установява, когато мрежата е разгърната и остава непроменена, освен когато се промени структурата на мрежата.
В симулатора е приложен алгоритъмът на авторите за формиране на енергийно балансирана мрежа ZigBee въз основа на приоритети с дървовидна топология. Топологията ZigBee се състои от един координатор (коренът на дървото), множество маршрутизатори (клонове) и крайни устройства (листа). В този алгоритъм методът на ценообразуване се използва за постигане на целта. В алгоритъма се приема, че рутерите служат само за изграждане на топологията и не функционират като крайни устройства. Всеки рутер и крайно устройство имат готовност да плащат стойност - приоритет за крайните устройства и ниво на енергия за рутерите. Координаторът и маршрутизаторите имат стойност на таксуване - цена, която трябва да бъде платена от крайните възли, за да се свържат с тях. Следователно, колкото по-висока е стойността на готовността за плащане, толкова по-висок е приоритетът на крайното устройство. При рутерите случаят е подобен, колкото по-висока е стойността на готовността за плащане, толкова повече енергия имат.
Симулаторът има модулна архитектура. Симулирането на ZigBee мрежа изисква работа през два основни прозореца. Единият от тях за добавяне на параметри за координатора, а другият за маршрутизаторите и крайните възли в мрежата.
Визуализацията на топологиите от проведените експерименти за внедрения алгоритъм за формиране на йерархична топология в мрежата ZigBee показва, че с увеличаване на броя на рутерите дълбочината на йерархията в изграденото дърво се увеличава. По отношение на енергийния баланс, алгоритъмът за формиране на йерархията гарантира, че рутерите с повече енергия са подредени на по-ниско ниво (по-близо до координатора). Това осигурява по-добра енергийна ефективност на рутерите, тъй като повече устройства ще бъдат свързани към тези с повече енергия и по-малко устройства към тези с по -малко енергия.
Експериментите показват, че внедреният алгоритъм позволява изграждането на балансирана по отношение на енергийната ефективност йерархична дървесна топология.

Г.7.17 Yuri Dimitrov, Veneta Aleksieva, Hristo Valchanov, Comparative Analysis of Prototypes for Two Touch Finger Interfaces of Smartwatch, IOP Conference Series: Materials Science and Engineering, CIEES'20, vol. 1032 (2020), pp.1-4, doi:10.1088/1757-899X/1032/1/012019, ISSN: 1757-8981, https://iopscience.iop.org/article/10.1088/1757-899X/1032/1/012019

Докладът представя сравнително проучване на предложения прототип за сензорен интерфейс за смарт часовник, активиран и управляван с два пръста, с други два прототипа.
За да се наблюдават зоните на докосване на безела, е проектиран и отпечатан специален 3D модел. Той е възможно най-близо до истинския смарт часовник според неговия размер и форма.
Първата стъпка е да се изберат размерите на 3D модела. Диаметрите на истинските интелигентни часовници варират между 34,5 мм и 58 мм, но 73% от тях са между 42 мм и 46 мм. Височината на истинските интелигентни часовници зависи от вида на функциите, но варира между 10,9 мм и 16 мм. Почти 80% от тях са между 14 мм и 15 мм. Въз основа на тази статистика избраните размери на 3D модела са - диаметър: 44 мм, височина: 15 мм, ъгъл на безела: 45^0, ширина на безела: 4 мм; цвят: бял; материал: PLA.
Втората стъпка е да се оцени в кои два отделни и обособени сектора на безела на устройството е възможно да се регистрират докосвания, за да се активира интерфейсът на устройството и да се извършат допълнителни действия с интерфейса. Подробните резултати от тази оценка са представени от авторите в други изследвания.
Третата стъпка е да се активират някои функции с този прототип на сензорен безел и да се сравни неговата функционалност с прототип с бутони. Авторите са направили това сравнение в други изследвания и основният извод е: прототипът с докосване надминава прототипа с бутони в скоростта на операциите, особено когато наборът от интерфейсни команди е по-дълъг.
Последната стъпка е да се оцени прототипът в сравнение с подобни прототипи. За да се направи сравнителен анализ на авторския прототип с други, се използват еднакви критерии за оценка. В прототипа на Oakley физическият контакт с безела на устройството е неразделна част от по-голямата част от входовете - само 17% са с два пръста. Участниците също предпочитат доминиращите си ръце и използването на палеца и показалеца си. Авторите публикуват резултатите за осем ординални посоки, но в това сравнение се използват само резултати за съвпадащите посоки с предлагания прототип. В прототипа на Yeo авторите използват няколко пръста, за да преместят целия прототип, който е с типичен размер на интелигентния часовник. Те показват, че техният прототип е конкурентен с търговските интелигентни часовници с този размер, като входните събития се генерират отзивчиво (55-61ms) и точно.
Експерименталните данни за прототипа на авторите са представени в сравнение с прототипа на Oakley и прототипа на Yeo в таблица. Според резултатите, получени за комплексната оценка, предложеният от авторите прототип е по-добър от двата други. Основното предимство на предложения прототип е неговият стандартен размер и по-малко време на докосване при дълга последователност от докосвания.

Г.7.18 Y. Dimitrov, V. Aleksieva and H. Valchanov, "Method for Body Pose Recognition based on Two-Finger Touch Bezel on Wearable Device", 2021 17th Conference on Electrical Machines, Drives and Power Systems (ELMA), 2021, pp. 1-5, doi: 10.1109/ELMA52514.2021.9503001.

Целта на представеното изследване е да предложи метод за разпознаване на позата на потребителя (легнал, седнал, изправен), когато събужда носимо устройство от режим на заспиване, и интерфейса на устройството да се активира, за да визуализира информацията и да изпълни някои команди при умишлено докосване на безела на носимо устройство с два пръста от потребителя.
Ако позата бъде успешно разпозната при активиране на интерфейса на носимо устройство, ще бъде предложен бърз достъп до функции и приложения в контекста на позата (тези, които най-вероятно ще бъдат изпълнени от потребителя). Това ще намали времето за активен режим на устройството, което ще удължи периода между две зареждания на батерията му. Разпознаването на позата на тялото може да се комбинира с други фактори, като времето през деня - например в легнало положение вечер, да се предложи бърз достъп до някои функции и приложения, а сутрин, отново в същото положение, да се предложи бърз достъп на потребителя до други функции/приложения. Друг фактор може да бъде предишна поза и/или дейност - например в изправено положение веднага след ставане, да предложи бърз достъп до някои функции/приложения, и в същата поза, но след дълъг период, да предложи други. По този начин разпознаването на пози при активиране на интерфейса на носимо устройство ще намали времето за работа с него, което ще доведе до по-дълъг период между две зареждания на батерията му. Умишлено докосване на панела за активиране на устройството с два пръста не може да бъде разпознато от друго действие и не може да възникне нежелано активиране на устройството.
Разпознаването на позата на тялото на потребителя се основава на относителната разлика в позицията на пръстите на безела при активиране на интерфейса от него в различните позиции на тялото му при използване на носимо устройство. Поради тази причина не е необходимо да се измерват ъглите в една и съща позиция за различни потребители, както и да се определят конкретни области на безела, за да се определи позицията на тялото. Достатъчно е всяко устройство / потребител да установи (след като започне да използва устройството) различните области на контакт при активиране на интерфейса и въз основа на тези различия да предвиди в каква позиция тялото на потребителя е най-вероятно в момента на активиране интерфейса.
Експерименталната група се състои от 10 души, всички с водеща дясна ръка, всички участващи доброволно в експеримента. Направени са 300 опита - по 100 за всяка позиция на тялото.
Въз основа на резултатите от експерименталните проучвания може да се предположи, че предложеният метод за определяне на позата на тялото на потребителя на носимо устройство въз основа на местоположението на пръстите на водещата му ръка върху сензорна рамка на носимо устройство, е ефективен и приложим.

Г.7.19 A. Haka, V. Aleksieva and H. Valchanov, "A Comparison Study of Decisions for Computer Network Laboratory in Distant Learning Education", 2021 17th Conference on Electrical Machines, Drives and Power Systems (ELMA), 2021, pp. 1-4, doi: 10.1109/ELMA52514.2021.9503059.

Този доклад представя сравнителен анализ на изследваните решения за дистанционно обучение по учебни дисциплини, преподаващи компютърни мрежи. Задължителната социална изолация по време на пандемия поставя нови предизвикателства пред образователната система. Необходимостта бързо да се премине към отдалечена форма на обучение наложи използването на различен образователен подход. По време на пълния lockdown бяха използвани и изследвани три подхода за обучение по дисциплини, свързани с преподаване на компютърни мрежи - симулационни продукти, реална компютърна мрежа с отдалечен достъп и виртуална компютърна мрежа с отдалечен достъп:

1) Да се използват симулационни продукти като Packet tracer, GNS3 и др.

2) В катедра „Компютърни науки и технологии“ към Технически университет-Варна е разработена от авторите реална компютърна мрежова лаборатория с отдалечен достъп. Отдалеченият достъп се осъществява чрез уеб система за управление, разработена от авторите. Citrix XenServer е избран за платформа за виртуализация, която има висока производителност, лесна поддръжка и е безплатна за използване. Основната идея на лабораторния дизайн е да се създаде snapshot на виртуалната машина (за съответната операционна система) за всеки от компютрите, като се използва възможността за snapshot на Xen.

3) За да се постигне висока гъвкавост и да се избегнат някои недостатъци на предишното решение, е внедрена експериментална виртуална инфраструктура. Базирана е на две сървърни машини Sun Fire Z20, свързани към 1G Ethernet мрежа и използващи VMware ESXi. Изборът на VMware Infrastructure 3 е продиктуван от възможностите му за многопроцесорна поддръжка, динамично балансиране и разпределение на ресурси между виртуални машини, както и мигриране на виртуални машини между отделни сървъри, без да се прекъсва тяхната работа. Въз основа на виртуалната инфраструктура бяха пуснати редица виртуални машини със съответни операционни системи. Виртуалната инфраструктура може да бъде достъпна със софтуера VMware vSphere Client.

Целта на изследването е да се оцени кое решение е най-подходящо за дистанционно обучение на студенти по дисциплини, свързани с компютърни мрежи. Разработена е система от критерии за оценка на горепосочените решения, съобразно предизвикателствата в онлайн обучението.

Сравнението се основава на предложена от авторите система от критерии, съобразена с предизвикателствата на дистанционното обучение. За да се осигури обективност при сравнението, е направена комплексна оценка на разглежданите подходи, базирана на комплексна аритметична оценка. Според резултатите от средна аритметична оценка най-подходящото решение за дистанционно обучение се определя решението с използване на виртуална мрежова инфраструктура.

Г.7.20 Haka, A., Yordanov, Y., Aleksieva, V., Valchanov, H. Study of Received Signal Strength Indicator values of Bluetooth Low Energy in Test Environment and Simulation, ICAI 2022, pp. 282–286, ISBN 978-166547625-6, DOI 10.1109/ICAI55857.2022.9960009

Този доклад представя изследване на RSSI стойностите при BLE технология в реална мрежа и симулация. Изследването цели сравняване на получените резултати при реална среда и симулация за формулиране на препоръки за ситуациите, при които може да се използва оборудване от специфичен производител и симулаторът. Разглежда се влиянието на броя на крайните възли и разстоянието между тях и централния върху силата на получения сигнал в реална среда, както и при симулация, за определяне достоверността на симулираните стойности и приложимостта им при изследване и обучение.
С експериментална цел, за изследване на RSSI стойностите при BLE технология в реална среда са свързани две BLE мрежи с крайни възли от два различни производителя – TI и Arduino. Компонентите при едната BLE мрежа са: Raspberry Pi 4 Model B с вграден BLE приемо-предавател и сензорни възли CC2650STK на фирмата TI. Ролята на главно BLE устройство се изпълнява от Raspberry Pi 4 Model B платката, на която предварително е заредена операционна система Raspbian. Крайните сензорни възли CC2650STK предварително са програмирани за работа с BLE стандарта. Компонентите при другата BLE мрежа са: главно устойство Arduino nano 33 IoT и сензорни възли Arduino nano 33 BLE sense на фирмата Arduino. Главното BLE устройство се реализира с Arduino nano 33 IoT платка, която е конфигурирана да работи с програмата, описана с псевдокод Код 1. Няколко платки Arduino nano 33 BLE sense се използват като крайни сензорни възли.
Експериментите, получени при BLE мрежа с крайни устройства от производителя TI показват, че отчетените RSSI стойности основно се изменят в диапазона от -40dBm до -70dBm, като изключение има при експериментите с 4 и 5 едновременно свързани крайни възли. При тези експерименти отчетените RSSI стойности се завишават за възлите които са разположени най-отдалечено от Master устройството. Отчетените стойности са в диапазона приблизително от -72dBm до -85dBm.
Стойностите за RSSI при експериментите в BLE мрежа с устройства на производителя Arduino се изменят основно в диапазона от -80dBm до -95dBm, като изключение има при експериментите с 5 и 6 едновременно свързани крайни възли. При тези експерименти отчетените RSSI стойности се завишават за възлите които са разположени след 3 метра от Master устройството. Отчетените стойности са в диапазона приблизително от -80dBm до -105dBm.
Реализирано е сравнение между проведените експерименти при еднакви условия. Въз основа на представените резултати са формулирани препоръки за случаите, при които може да се използват разглежданите крайни възли и симулаторът.

Г.7.21 Haka, A., Dinev, D., Aleksieva, V., Valchanov, H. Internet of Things Sensor Data Storing Systems for Educational Purposes, CIEES’22, 2022, Electronic ISBN:978-1-6654-9149-5, DOI: 10.1109/CIEES55704.2022.9990805

Този доклад представя две предложени и разработени системи за съхранение на данни от сензорни LoRa и ZigBee мрежи, които имат широко приложение в областта на IoT. Системите са създадени в катедра „Компютърни науки и техника” на Технически университет – Варна за използване в обучението. Чрез разработването на уеб интерфейс те осигуряват дистанционно наблюдение на данните, съхранявани в нерелационна база данни. Системите позволяват на студентите да се запознаят с възможностите за конфигуриране и работа на два от най-използваните стандарти за IoT и домашна автоматизация. Те също така позволяват на студентите да изучават работата на протокола MQTT. Освен това те дават възможност на студентите да утвърдят и придобият нови знания и умения в областта на управлението на бази данни и програмирането.
За целите на обучението в катедра КНТ, към Технически университет - Варна е разработена LoRa система за съхранение на сензорни данни за IoT с Уеб-базиран интерфейс за работа с възможност за отдалечен достъп. За конфигуриране на системата не е необходимо генериране на специални идентификатори. Системата се състои от три основни компонента Dragino LG01-S - Single Channel LoRa IoT Gateway за управление на мрежата и получаване на данни от сензорни възли, MQTT сървър за получаване и препращане на данни от LoRa Gateway и MongoDB база данни за съхраняване на получената информация. На LoRa Gateway е необходимо конфигуриране за препращане на получените съобщения от сензорите към MQTT сървър. След това MQTT съобщенията от съответния канал се предават на MongoDB базата данни, където се съхраняват в съответната колекция. Предаването на информация от MQTT сървър към MongoDB се осигурява с Python скриптове. Разработеният Уеб интерфейс осигурява подходяща визуализация на съхранената информация и извеждане на статистически извадки на база отделните характеристики и отрязъци от време на работа, както и създаване на модел на работа на устройствата в изследваната среда.
Представени са предимства и недостатъци на съществуващи решения за съхранение на данни при ZigBee, както и на разработеното. Според представената информация разработеното решение включва в себе си част от предимствата на съществуващите и преодолява повечето от отбелязаните им недостатъци. Разработеното решение има недостатъци във възможностите за осигуряване на среда за работа на множество потребители, ограничено хранилище за съхранение на информацията и споделяне на информация. Разработеното решение позволява работа с различни технологии за IoT, а представените недостатъци може да се преодолеят с подобряване на системата. Това показва, че разработената система е подходяща за целите на обучението в университетска среда.

Г.7.22 Haka, A., Dinev, D., Aleksieva, V., Valchanov, H. A Study of ZigBee Networks in Experimental Environment and Simulation, CIEES’22, 2022, Electronic ISBN:978-1-6654-9149-5, DOI: 10.1109/CIEES55704.2022.9990742

Този доклад представя изследване на параметрите End_to_End Delay, Throughput и Packet Loss Ratio (PLR), които влияят на QoS при една от най-често използваните IoT технологии – ZigBee. Направените изследвания представят резултати за разглежданите параметри на база симулация и реална ZigBee мрежа с крайни възли на производителите Texas Instruments (TI) и Sonoff. На база изчислените стойности е направено сравнение между резултатите, целящо формулиране на препоръки за избор на устройства за изграждане на ZegBee мрежа, според специфичните QoS изисквания в мрежата, както и определяне възможностите за приложение на разглеждания симулационен продукт.
Експерименталните изследвания при разглежданите ZigBee устройства са извършени при различен брой едновременно свързани в мрежата статични крайни възли (2, 4 и 6). При всеки експеримент крайните възли са разположени на различно разстояние от главното устройство (1m, 2m, 3m, 4m и 5m), като при всяко от разстоянията от всеки възел са изпратени 5, 10, 15 и 20 пакета. Топологията на свързване за проведените експерименти е звезда.
ZigBee мрежата с крайни възли на производителя TI е изградена с BeagleBone Black (BBB) - BBB01-SC-505 платка, инсталирана със софтуер Z-Stack Linux Gateway, за работа като ZigBee координатор. Към BBB платката е свързан приемо-предавател TI CC2531EMK, конфигуриран да предава и приема ZigBee сигнали. Крайните устройства са TI CC2650STK, конфигурирани за работа като ZigBee възли.
ZigBee мрежата с крайни възли на производителя Sonoff е изградена със софтуер ZigBee2MQTT, инсталиран на Windows PC, който реализира ZigBee координатор. Към Windows PC е свързан ZigBee приемо-предавател TI CC2531EMK за комуникация с крайните възли, също така е инсталиран Mosquitto сървър и Nodejs за прочитане данните от крайни възли. Крайните устройства са Sonoff Door Sensor и Sonoff Temperature and Humidity Sensor.
ZigBee мрежата през симулатора е изградена изцяло виртуално с параметри за координатор и крайни възли, базирани на публикуваните стандарти.
На база проведените експерименти и обобщените резултати може да се формулират следните препоръки:

- За ZigBee приложения, при които се изисква значително ниски и постоянни End_to_End Delay стойности е по-добре да се използват TI устройства в мрежата;
- За ZigBee приложения, при които се изискват по-високи Throughput стойности е по-добре да се използват TI устройства в мрежата;
- За ZigBee приложения, при които се изисква поддържане на ниски PLR стойности може да се използват както TI, така и Sonoff устройства в мрежата;

За изследване посоката и тенденцията на промяна на изследваните QoS параметри при различни ситуации, може да се използва разглежданата симулационна среда, която е подходяща и за използване при обучение както в присъствена, така и в отдалечена форма.

Г.8. Публикации в нереферирани списания с научно рецензиране

Г.8.1 Вълчанов Х. Виртуализирана мрежова лаборатория. Национална конференция по е-обучение във висшите училища, Русе, 2014, 159-164.

Изграждането и поддържането на лаборатории за обучение по компютърни мрежи и Интернет технологии е практически сложна задача. Потребителите трябва да могат да изграждат различни мрежови топологии, както и да имат пълен административен контрол върху устройствата. Същевременно, преди провеждането на практически занимания и изследвания, студентите трябва да изградят съответната опитна постановка. Това изисква време, а от друга страна, са възможни грешни свързвания, които могат да доведат до повреда на оборудването.
Същността на предлагания подход е използването на наличното оборудване в съществуваща лаборатория по компютърни мрежи, като всеки от компютрите е конфигуриран за изпълнението на определена роля според съответен мрежови сценарий. За целта се използва една характерна особеност на виртуализационните платформи – т.н. snapshot, която представлява актуално копие на моментното състояние на виртуалната машина. Актуалното копие съдържа IP конфигурация и набор от стартирани услуги, които определят ролята на съответната машина във виртуалната инфраструктура. Посредством актуалните копия на виртуалните машини се реализира бързо превключване между различни инфраструктури, като в рамките на малък интервал от време се изгражда заявената функционираща виртуална инфраструктура.
Използваната платформа за виртуализация е KVM (Kernel-based Virtual Machine). Изборът на KVM е продиктуван от следните съображения. KVM е тясно интегриран с емулатора на виртуален хардуер QEMU (Quick EMUlator), който осигурява възможности за създаване на актуални копия на състоянието на машините. Управлението е реализирано посредством web интерфейс, базиран на PHP и AJAX. За ускоряване на процеса на преконфигуриране се използват паралелни AJAX заявки към web сървъра.
Представени са изследвания за действията, необходими за изграждането и пускане в действие на определена мрежова топология. Резултатите показват, че продължителност на процеса по пускане в действие на функционираща мрежова инфраструктура за конкретна задача е около 46 минути. Това представлява почти половината от предвиденото за лабораторни занятия време. Като резултат, учебният материал не може да бъде усвоен от студентите в необходимия обем – това рефлектира в намаляване качеството на процеса на обучение.
Внедряването на виртуализираната мрежова лаборатория дава възможност за елиминирането на тези проблеми. Изграждането на конкретна мрежова топология се извършва в рамките на 1 минута Студентите получават изградена и функционираща инфраструктура, което позволява те да насочат усилията си директно за решаването на конкретната практическа задача. Това рефлектира както в успеваемостта при решаването на задачите, така и в повишаване качеството на процеса на обучение на студентите.

Г.8.2 Вълчанов Х. Прехвърляне на мултимедиен трафик през WAN мрежи. Proc. of Int. Conf. Automatics and Informatics'14, Sofia, 2014, pp.I-167 – I-170. ISSN 1313-1850

Многопотоковото предаване на транспортно ниво е възможността на транспортните протоколи да подържат различни потоци данни, като за всеки поток се осигурява независима последователност на доставяне на данните. Протоколът Stream Control Transmission Protocol (SCTP) е стандартен надежден транспортен протокол, осигуряващ многопотоково предаване на данни.
В представения доклад се предлага подход, който се състои в използване на съществуващите софтуерни решения (HTTP сървъри и клиенти), функциониращи на базата на протокола TCP, и предаване на данните през WAN мрежи посредством протокола SCTP. Мотивацията за предлаганото решение е базирана на факта, че сървърите и клиентите се намират в локални мрежи, за които е характерна висока пропускателна способност. Тези мрежи са свързани типично посредством нискоскоростни WAN технологии, имащи в порядък по-малка пропускателна способност. С предлагания подход могат да се използват пълните възможности на SCTP за моногопотоков обмен.
В доклада са представени са някои особености на реализацията на TCP/SCTP прокси сървър. Вътрешната структура на прокси сървъра включва четири отделни модула:

- Комуникационният управляващ модул поддържа информация за определяне на коректния сокет, по който получаваният пакет се доставя.
- Управляващият модул на TCP сесията е базиран на системната библиотека на Linux. Той създава и преустановява на сесии и контролира на обмена на данните с откриване на загубени пакети и отчитане на задръстванията в мрежата.
- Управляващият модул на SCTP сесията е базиран на библиотеката sctplib. Той подрежда пакетите в рамките на поток – за всеки отделен поток модулът наблюдава поредността на получаване на данните. Загуба на данни в даден поток не се отразява на останалите потоци. Същевременно осигурява създаване и преустановяване на асоциации – включва механизми за създаване, нормално затваряне и прекъсване на асоциация.

Изградена е тестова инфраструктура, включваща WAN връзки, върху която са проведени групи експерименти. Извършени са две групи експерименти – файлов обмен и зареждане на web страници. Тестовете са направени при различни скорости на серийните връзки между маршрутизаторите - 64000 bps, 115200 bps и 128000 bps. При експериментите със зареждане на web страница се наблюдават по-добри времена на отговор при използване на протокола SCTP. Тези резултати се получават благодарение на общия механизъм на SCTP за управление на потока данни. При изтегляне на даннов файл, механизмът за управление на SCTP потоците не е ефективен, тъй като се използва само един поток. В противоположност, протоколът TCP се нуждае единствено от единична сесия за предаването на файла.
Цел на бъдеща работа е разработването на многонишкова архитектура на прокси сървъра.

Г.8.3 Valchanov H., M. Angelov. Improving Performance of Multimedia Web Transfer over WAN Connections. Proc.of the ICEST 2014, Nis, Serbia, v.1, 27-30. ISBN 978-86-6125-108-5.

Докладът представя подход за подобряване на производителността на предаване на данни между мрежи, базирани на WAN технологии. Показани са архитектурата и вътрешните подробности на SCTP уеб прокси сървъра. Основната идея на предложения подход е, че представеният TCP / SCTP прокси сървър ще работи като интерфейс между TCP и SCTP протоколите, позволявайки на уеб браузърите и сървърите да се възползват от възможностите на SCTP, без да се налага да променят своя код. Тъй като целта на настоящото изследване е проучване на възможностите на подхода, структурата на прокси сървъра е опростена, като предоставя само базова функционалност, необходима за провеждане на проучването.

Функционалността на прокси сървъра е реализирана на основарта на модел с двоен стек. Двойният стек използва два транспортни протокола – TCP и SCTP, като по този начин позволява лесна интеграция на прокси сървъра в TCP/IP инфраструктурите. Когато се получи TCP заявка от клиент (уеб браузър), проксито действа като TCP сървър. Когато препрати тази заявка към друг SCTP прокси, той работи като SCTP клиент. По същия начин, при получаване на отговор от TCP сървър (уеб сървър), проксито действа като TCP клиент, а при връщане на отговора на SCTP прокси работи като SCTP сървър. Компонентът Proxy Core изпълнява основната функционалност на прокси сървър.

Анализът на производителността на предложения подход е направен върху мрежова инфраструктура, представляваща бавни WAN връзки. Тестовата среда включва Cisco рутери 2901, VLAN Cisco Catalyst 2960 комутатори, компютри HP Desktop 500B CPU Intel Core Duo E5800 3, 2 GHz с 2G RAM. Платформата на уеб сървъра е базирана на Slackware Linux 2.6 и gcc 4.4.3. Като клиент се използва операционна система Windows 7 и браузър Google Chrome 28.0.1500.

Проведени са две тестови групи – прехвърляне на файлове и зареждане на уеб страници. Проведени са експерименти при различни скорости на серийните връзки между рутерите - 64000 bps, 115200 bps и 128000 bps.

Получените резултати показват, че предложеният подход е напълно функционален и приложим за пренос на мултимедийни данни през мрежи с ниска честотна лента.

Целта на бъдещата работа е да се разработи многонишкова архитектура на прокси сървъра. Това ще му позволи да обслужва едновременно множество клиенти. Друга бъдеща работа е да се добавят възможности за кеширане на данни и филтриране на трафика.

Г.8.4 Вълчанов Х. Подход за мултимедиен Web трансфер през нискоскоростни глобални мрежи. Proc. of UNITECH'14, Gabrovo, 2014, pp.II-213 – II-218. ISSN 1313-230X

Използването на многопотоковия протокол SCTP като транспортен протокол за HTTP. може да реши редица проблеми при използвания в момента модел на предаване на мултимедийни документи в WAN инфраструктури. Поради факта, че мултимедийните документи се състоят от обекти от различни типове и размер, многопотоковото предаване позволява тяхното изпращане в частична подредба, вместо в стриктна последователност. Като резултат се подобрява визуализацията на страниците при зареждане в браузъра. В същото време, транспортът се осъществява в рамките на една асоциация, като по този начин всички потоци използват общ механизъм за управление на обмена на данни. Това от своя страна, значително редуцира системните разходи на транспортно ниво.

В представения доклад се предлага различен подход, който се състои в използване на съществуващите софтуерни решения (HTTP сървъри и клиенти), функциониращи на базата на протокола TCP, и предаване на данните през WAN мрежи посредством протокола SCTP.

В доклада е представен TRCP/SCTP прокси сървър с модулна архитектура.

Целта на проведените експерименти е да се изследва поведението на двата транспортни протокола в мрежови инфраструктури, изградени на базата на WAN технологии. Тези видове технологии предоставят пропускателна способност в порядъци по-малка от технологиите, използвани при локалните мрежи. Това се явява важен фактор, който глобално влияе върху времето за отговор на web сървърите.

Изградена е експериментална мрежова инфраструктура, като между маршрутизаторите са изградени серийни връзки, базирани на протокола High-Level Data Link Control (HDLC). Посредством тяхното конфигуриране с различни скорости е симулирана преносна WAN среда с различна пропускателна способност. Експериментите са проведени в две направления:

- директна комуникация между web клиент и сървър, базирана на протокола TCP;
- индиректна комуникация между web клиент и сървър, реализирана през TCP/SCTP прокси сървър.

Получените резултати показват, че при експериментите със зареждане на web страница се наблюдават по-добри времена на отговор при използване на протокола SCTP. Зареждането на web страница се извършва в рамките на единична SCTP сесия (в контраст с TCP, при който за всеки ресурс се създава нова сесия).

При изтегляне на даннов файл, механизмът за управление на SCTP потоците не е ефективен, тъй като се използва само един поток. В допълнение следва да се отбележи, че и двата транспортни протоколи осигуряват надеждно доставяне на данните без загуба на информация.

Г.8.5. V.Aleksieva, H.Valchanov, T. Dlugosz, R. Wrobel. Real efficiency of SoHo routers with alternative software. Telecommunication review+Telecommunication news Tele-Radio-Electronics, Poland, v.1, pp.10-12, 2014, ISSN 1230-3496.

Статията представя анализ на ефективността на три популярни модели на SoHo (Small Office Home Office) рутери. Авторите представят преглед на развитието на SoHo мрежи в България и Полша. Тествана е ефективността на тези три модела въз основа на честотната лента за UDP и TCP протокол в три топологии. Резултатите са представени в таблици и графики, което позволява сравнението им. Скоростта на трансфер на данни е един параметър, който беше проучен. Не само цената и допълнителните възможности имат влияние върху избора на SoHo рутери. Най-важните фактори за избор са топологията и видът на трансферираните данни. В резултат на проучването, най-продуктивен е WR1043ND маршрутизатор на TP-Link.

Г.8.6 H.Valchanov, S.Andreev. Muti-threaded user and kernel-space library. Proc.. of the ICEST2015, Sofia, 2015, pp.208-211

Развитието на технологиите и голямата гама от възможности, предлагани от съвременния хардуер, позволяват използването на специализирани високопроизводителни подходи за внедряване на различни софтуерни системи и алгоритми. Един от най-използваните и ефективни подходи е създаването на многонишков софтуер, работещ паралелно на множество процесори. Този доклад представя особеностите на реализацията на многонишкова библиотека под Linux, което позволява работа както в потребителски (user-space), така и в режим на ядрото (kernel-space).

В потребителски режим всяка нишка е представена от специална структура (Thread Control Block - TCB), съдържаща необходимата информация за нейното управление. Тази информация се използва от диспечера за планиране и превключване на контекста. Диспечерът се грижи за управлението на събитията, което е много важно за правилното функциониране на библиотеката. Диспечерът се извиква всеки път, когато една нишка предостави процесора на друга нишка или когато се блокира заключи автоматично към събитие.

Реализирането на нишките в този режим се основава на системно извикване clone(). За ядрото на Linux една нишка се съхранява в същата структура, която се използва за отделен процес. Въпреки, че основната информация се съхранява в структурите на операционната система, все още е необходимо да се поддържат данни за нишката и в потребителския контекст. Такава информация, например е за началната функция и нейния аргумент. Поддържането на тази дублирана информация позволява по-просто изпълнение на някои библиотечни функции.

Проведени са експериментални изследвания за тестване на ефикасността на двете реализация чрез изпълнение на изчислителни задачи и задачи със интензивен вход/изход. Направено е сравнение с библиотеката pthreads.

При изчсилителните задаяи, като резултат, pthreads и реализацията в kernel-space се справят най-добре с теста. Резултатите са близки, но предимството е за библиотеката pthreads. От този тест се вижда големият недостатък на user-space библиотеките – те могат да работят само на един процесор.

При тестовете за вход/изход библиотеката pthreads и реазлизацията в kernel-space показват една и съща производителност с лек превес на реализацията на kernel-space.

Като цяло, резултатите показват, че за представената библиотека се постига идентична и в конкретни случаи висока производителност. Разработената библиотека е относително малка, което води до по-бързо компилиране. Кодът е написан по прост начин, което позволява използването на библиотеката за изучаване на многонишковото програмиране.

Г.8.7 Х.Вълчанов. Хибридна многонишкова библиотека. Сб. на международна конференция "Автоматика и информатика'2015, София, 2015. 157-160. ISSN 1313-1850

Многонишковото програмиране е широко използван съвременен подход за повишаване на производителността на изчисленията чрез въвеждане на паралелизъм при изпълнението на програмите. В настоящия доклад са представени някои аспекти на реализацията на хибридна многонишкова библиотека под Linux, която дава възможност за функциониране както в потребителски режим (user-space), така и в системен (kernel-space). Библиотеката предоставя приложен потребителски интерфейс, който е максимално идентичен за реализациите в двата режима.

Представена е реализация в user-space режим при която е използвана non-preemptive схема на диспечеризиране. Показани са различните състояния на нишките в този режим. Неблокиращите read/write операции, заспиване на нишка за определено време, операциите със синхронизиращите примитиви и функционалността за свързване на една нишка с друга са изцяло базирани на системата за събития в Linux.

При реализацията в kernel-space режим планирането и диспечеризирането на нишките се използват системните средства, предоставени на ниво операционна система. В текущата реализация на библиотеката са разработени два типа примитиви за синхронизация – spinlocks и mutexes. При spinlocks синхронизация е базирана на активно изчакване, при което се заемат процесорни цикли. От гледна точка на ефективност този подход не е желателен, но реализацията е изключително опростена. Примитивите за синхронизация mutexes са реализирани чрез смесен подход, с което се цели постигане на по голяма ефективност. Първоначално се прави опит за кратко активно изчакване. Ако след това mutex е все още зает, се преминава към блокиране на изпълнението на нишката от ядрото. За целта се използва предоставяното от Linux средство за базово заключване на достъпа Fast User-Space Mutex (futex).

Тестването на многонишковата библиотека е направено по отношение на двата основни вида натоварване на системата:

- С процеси, ограничени по процесорно време. Това са процеси, които изпълняват голям обем изчисления.
- С процеси, ограничени по входно-изходни операции. Това са процеси, които изпълняват интензивно системни извиквания към операционната система.

Първият тест включва умножение на матрици, като пример на задача, съдържаща множество изчисления. Библиотеката pthreads и kernel-space реализациите имат най-добри резултати.

Третият тест цели оценка на ефективността на синхронизиращите примитиви при достъп до общ ресурс. В цикъл от 1000000 итерации се заключва и се отключва mutex. Създават се 5 отделни нишки, всяка от които изпълнява описаното действие. Резултатите показват, че реализация на mutex в библиотеката pthreads е по-бавна и изисква средно 615154μs. Версията в kernel-space библиотеката е около 1.5 пъти по-бърза, благодарение на по-простата и ефективна структура на организацията на нишките.

Г.8.8 Вълчанов Х., Д.Тодоров, Система за индексирано търсене в локална Windows мрежа. Сб. Научна конференция 2015, РУ, 57-61, ISSN 1311-3321

В докладът е представена архитектурата на разпределена система за индексиране и търсене в локална Windows мрежа. Системата индексира освен имена на файлове и тяхното съдържание. Процесът на индексиране може да се изпълни както в оперативната, така и в дисковата памет. Системата позволява при търсене задаването на сложни булеви заявки. Достъпът до функционалността на системата е организиран на базата на потребителски групи.

Архитектурата на системата за индексирано търсене в локална Windows мрежа има разпределена организация. Тя се състои от множество независими услуги, стартирани върху отделни машини от мрежата. Услугата се състои от следните основни компоненти: подсистема за индексиране, подсистема за търсене и комуникационна подсистема. Всяка машина от мрежата съхранява само своите индекси. По този начин отпада нуждата от използване на сървърна машина, която да съхранява всички индекси на участниците в локалната мрежа. Като резултат се намалява обема на съобщенията през мрежата.

Приложената стратегия позволява всеки компонент от локалната мрежа сам да менажира своите индекси. Всяка от услугите може да приема локални заявки за търсене, както и заявки от мрежата. Обменът на информация между отделните компоненти през мрежата се извършва от комуникационната подсистема.

Експерименталните изследвания са направени в локална мрежа със свързани Windows машини. Проведени са две групи тестове - за индексиране на зададена директория с обем 2,12 Gb и търсене на заявки на локалната машина.

Представените резултати показват, че при индексирането Натоварването на системата е много голямо при Microsoft индексирането и индексирането в оперативната памет. При индексирането в оперативната памет всички индекси се съхраняват в оперативната памет, което води до високото натоварване, докато за Microsoft няма сведения, как изграждат индексите.

При обработването на заявки резултатите показват, че обработка на заявка с маска в ОП (с дървовидна структура) се извършва два пъти по-бързо отколкото при дискова памет (чрез многократно зареждане на векторен буфер), а с най-голямо време е Microsoft търсенето.

Г.8.9 Todorov D., H. Valchanov. Multicast Indexing and Search System in Local Network. Proc. of UNITEH’15, Gabrovo, 2015, pp.II-269 - II-274. ISSN 1313-230X

В докладът е представена системата за индексиране и търсене с разпределена архитектура, Тя се състои от множество търсещи машини (ТМ) които са инсталирани като услуги върху всички машини от локалната мрежа, участващи в процеса на търсене на информация. Всяка ТМ функционира независимо от останалите, като индексира информацията в локалната машина. Заявките за търсене могат да бъдат издавани локално (от потребителя на локалната машина) или да бъдат изпращани към всички ядра през мрежата. Обменът на информация между ТМ е реализиран на базата на мултикаст. Всяка ТМ се състои от три основни компонента: за индексиране, за търсене и за комуникация.
Индексирането е базирано на т.н. инвертен индекс. Той представлява списък от открити термини (думи), като за всяка дума се изгражда списък от индексите на файловете, в който се съдържа този термин. Характерно е, че всяка дума е уникална, като се използва ключ при обработката на заявките за търсене. За всяка дума се поддържа позицията и честотата на срещане във всеки файл. Това позволява да се извършва търсене по фрази, както и сортиране на получените резултати.
Реализацията на комуникацията в представената система е базирана на Winsock2 мултикаст. Естеството на обменяните данни (думи, фрази за търсене, резултати от търсенето под формата на пътеки към файлове и самите файлове) изисква те да бъдат доставяни надеждно. Заложеният в Winsock2 мултикаст протокол за надеждна доставка е Pragmatic General Multicast (PGM).
Проведени са изследвания с цел да се изследва функционалността на предлаганата система в реална локална мрежа и се сравни нейната ефективност с тази на вградената в Windows система за индексиране и търсене.
Експериментите са извършени в локална мрежа, включваща четири клиента и отделна машина за автентикатор. Компютрите са под управлението на Windows 7 Enterprise. Извършени са два типа тестове: индексиране и търсене по заявки:

- Индексиране. Тестват се три типа индексиране: индексиране в оперативната памет, индексиране в дисковата памет и Windows индексиране.
- Изпълнение на заявки. Тестват се четири типа заявки:
  - търсене на термин – term;
  - търсене с маска – term*;
  - търсене на фраза – term1 term2;
  - логическо търсене от типа term1 AND term2 OR term2

При експериментите с индексиране в ОП се наблюдават закономерно по-добри времена в сравнение с индексирането в дисковата памет, както и в порядък спрямо Microsoft. При обработката на заявка с маска в ОП (с дървовидна структура), тя се извършва два пъти по-бързо отколкото при дискова памет (чрез многократно зареждане на векторен буфер), а с най-голямо време е Microsoft търсенето. При търсене на единична дума и при логически изрази отново най-добри резултати дава търсене в ОП, алгоритъмът с векторен буфер е по-бавен, докато при Microsoft изпълнението е два пъти по-бавно от дисковото търсене.

Г.8.10 Николов В., Х. Вълчанов. Система за анализ и диагностика на цифрови изображения на кръвни проби. Компютърни науки и технологии, 2015 81-88, ISSN 1312-3335.

В статията е представена система за бърза диагностика на заболявания, засягащи състоянието на кръвните клетки като промени в големината, формата, оцветяването, наличие на включвания в тях и др. Представеното решение се базира на анализ на изображенията на кръвните клетки, чрез определяне на техните контури, чрез които се формира описание и се обучава невронна мрежа за разпознаване.

Тъй като изображенията на кръвните проби са с голям брой визуални обекти, които трябва автоматично да се анализират, се налага извличането само на определени характеристики от сегментите в изображението. За целта се съставят признаци, всеки от които представлява апроксимираща права линия на контура на кръвна клетка. Един признак се представя чрез вектор от пет елемента (x1, y1, x2, y2, sin, cos), като за тяхното определяне се изпълняват следните стъпки:

- цифрово представяне на изображението;
- определяне на контурите на визуалните обекти (червени кръвни клетки) и тяхното изтъняване;
- обхождане на контурите и определяне на контролни точки за обособяване на признаци;
- формиране на описание на обектите в изображението чрез описание на признаците.

След формиране на описанието на контурите на изображенията се пристъпва към обучението на невронната мрежа. Тя е от тип многослоен персептрон и се обучава по широко използвания алгоритъм с обратно разпространение на грешката.

Структурата на системата се състои от административен модул и множество от клиентски модули. В административния модул се съдържа информация за заболявания, която се предоставя към клиентските модули с помощта на уеб услуги. Информацията се изгражда, чрез предварително структурирани и групирани според заболяванията изображения на кръвни клетки. Класифицирането се извършва от експерти, медицински лица, които предоставят примери за използване.

Клиентските модули могат да инициират заявка за най-актуалните дефиниции на заболявания и да извършат класифициране на подадено изображение. В резултат от заявката, главният модул предоставя обучена невронна мрежа във вид на XML дърво, която се построява в паметта при клиента. XML информацията съдържа архитектурата на невронната мрежа (брой слоеве, брой неврони) и самите тегла на връзките. Функциите на клиентския модул позволяват единствено диагностика посредством определяне на категорията на представеното на входа изображение.

Представената система използва конекционистки подход за класификация на изображения, представени във вид на матрици от числа, с използване на информацията за предварително формирани класове от изображения. Цел на бъдеща работа е интегрирането на представената система в цялостна инфраструктура за ранно известяване при епидемии.

Г.8.11 Вълчанов Х. Комуникационна среда за търсене в локална мрежа. Proc. of TechSys’16. Plovdiv, 2016. ISSN 1310 – 8271, II-199 – II-202..

В докладът е представена архитектурата на комуникационна подсистема на разпределена среда за индексиране и търсене в локална Windows мрежа. Системата за индексиране и търсене има разпределена архитектура. Тя се състои от множество търсещи машини които са инсталирани като услуги върху всички машини от локалната мрежа, участващи в процеса на търсене на информация. Системата позволява изпълнение на няколко типа заявки: търсене по дума, по част от дума (търсене с маска), търсене на фраза и сложни логически изрази.
Комуникационната подсистема осигурява обмена на информация между изпълняваните на отделни машини ТМ. Този обмен включва предаване на заявки за търсене към всички машини и изпращането на получения от търсенето резултат. Комуникацията в реализирана на Winsock2 мултикаст протокола PGM.
Комуникационната подсистема се състои няколко базови компонента:

- Communication kernel – ядро, осигуряващо базовата функционалност.
- User message queue - синхронизирана опашка от съобщения, които са предназначени да се предадат или обработят.
- Search result – буфер за съхраняване резултатите от търсенията.
- Hash – буфер за хеша, използван във фазите на процеса на автентикация.
- RX, TX port – PGM сокети за получаване и изпращане на съобщения по мрежата.

Мултикаст адресът, по който комуникира системата е фиксиран и общ за всички инстанции на приложението. В този му вид всеки потребител би участвал в търсенията. В една реална среда това е крайно недостатъчно. С цел разделяне на потребителите на подмножества и ограничаване на търсенията до рамките на даденото подмножество е въведен механизмът на групите. Всеки потребител може да е член на една или няколко групи. Реализирана е защита от прочитане на данните на база на концепцията за криптиране на мултикаст. За разлика от класическата схема, вместо да има един ключ за мултикаст групата, се въвежда по един ключ за всяка група от системата, наричан групов ключ, осигуряван от специален централизиран компонент – автентикатор.
Проведени са експериментални изследвания в локална мрежа с Windows машини. Извършени са два типа тестове: индексиране и търсене по заявки. Процесът на индексиране може да се изпълни както в оперативната, така и в дисковата памет. Направени са експериментални сравнения и оценки на двата метода за индексиране на информация, както и търсене с вградената във Windows 7 система за индексиране и търсене. Резултатите показват, че е постигнато по-добро бързодействие от съществуващата реализация във Windows.
Като насоки за бъдеща работа се предвижда разработване на хибриден индексиращ алгоритъм, както и автоматично генериране на ключове за автентикация.

Г..8.12 V.Aleksieva, H.Valchanov, M.Magdziak-Toklowicz, R.Wrobel, R.Wlostowski, Transmission of vibrations from the engine to the car body, Journal of KONES Powertrain and Transport,vol.23,No.4 2016, pp.17-23, ISSN:1231-4005

Вибрациите се превърнаха във важен фактор за превозните средства. Вибрационните тестове помагат да се идентифицира и след това да се настрои автомобилното превозно средство, за да се подобри здравината на конструкцията. Вибрационното изпитване често се извършва с помощта на лазерна доплерова виброметрия (LDV) - устройство, което се използва за безконтактно измерване на вибрациите на повърхността. Лазерният лъч се насочва от устройството към повърхността, която представлява интерес, а амплитудата и честотата на вибрациите се извличат от честотата на доплеровото изместване на отразения лазерен лъч поради движение на повърхността. Високите стойности на вибрации, предавани от двигателя, и начина, по който влияят значително върху каросерията на превозното средство и водача са изследвани.

Статията представя резултатите от изследванията, проведени върху превозни средства, задвижвани от три различни двигателя и обороти. Изпитанията бяха проведени на динамометър на двигателя при еднакви условия на околната среда. Два от двигателите бяха с искрово запалване, включително един с двигател с компресор и двигател със запалване под налягане.

Измерванията са направени с помощта на лазерна доплерова виброметрия, използваща бърза трансформация на Фурие. Полученият спектър се използва за по-нататъшен анализ за определяне на нивото на ускорение при различни честоти. Получените показания за бързо преобразуване на Фурие са използвани за начертаване на графики на честотното ускорение.

С увеличаването на скоростта на въртене на коляновия вал (и намаляване на продължителността на периода) се появяват допълнителни колебания при всички видове превозни средства (те се виждат ясно дори при най -ниската скорост на двигателя с компресор), но вибрационният сигнал е със стационарен характер.

Диаграмите показват недвусмислено, че амплитудата на вибрацията, независимо от целта на измерване, е най-голяма за превозното средство с двигател със запалване под налягане и най-ниска за превозното средство с двигател с искрово запалване (без налягане). В същото време колебанията и средните стойности на сигналите показват, че превозното средство с дизелов двигател е най - ергономично, докато превозното средство с двигател с искрово запалване със свръхкомпресор е най -малко ергономично.

Г.8.13 Алексиева Ю., Х. Вълчанов. Симулатор на Ботнет DoS атаки в мрежова среда. Сб. на международна конференция "Автоматика и информатика”, 2016, София, 2016. 163-166. ISSN 1313-1850

Един широко разпространен вариант на ботнет атаки е IRC ботнет. Има редица техники за откриване на ботнет атаки, но те нямат функционалността да предотвратят злонамерена ботнет дейност. Разработването на качествено оборудване за откриване и премахване на ботнет заплаха се нуждае от добра симулационна среда, която е безопасна и напълно отговаря на функционалността на ботнет DoS атаките. Докладът представя особеностите на реализацията на мрежов симулатор за DoS атаки, позволяваш създаване на различни атаки с различни параметри.

Архитектурата на представената симулационна среда има разпределен характер. Тя се състои от множество ботове, изпълняващи се на отделни машини и комуникиращи си през IRC канали. Всеки бот е изграден на модулен принцип. Използването на модули дава възможност за лесно последващо разширение на симулатора с нови компоненти, реализиращи допълнителни класове атаки.

Представената симулационна среда има за цел да симулира действието на зловредни мрежови атаки. Поради тази причина, начинът на превръщане на машините в ботове не е обект на изследване в представения доклад. Приема се, че ботнет мрежата е вече създадена – компютрите са били заразени. За да може да се симулира действието на IRC ботнет, всяка машина трябва да има инсталиран предварително бот. Заради възможността този софтуер да бъде използван за злоупотреба не е добавена функционалността той да се стартира автоматично при включване на машина.

Необходимостта от комбинативност по отношение на изпълнението на различните видове атаки изисква всеки вид атака да може да бъде изпълнявана автономно с максимални възможности за параметризация. Като параметри се задават имена/адреси на хостове, номера на портове, брой пакети, размер на съдържанието на пакет и др. Всяка атака може да се конфигурира с определена продължителност на времетраене. След всяко изпълнение на команда или атака, ботовете съобщават статуса си чрез персонално съобщение към потребителя. Всеки бот изпраща съобщение с отчет за това колко пакета е изпратил, колко време е продължила атаката и какъв е резултата от изпълнена команда.

Тестването на симулационната среда е направено по отношение на два основни типа мрежова инфраструктура- равнинна и маршрутизирана.

Тестовете са извършени в две насоки: за тестване на комуникацията между ботовете и за реализация на мрежови атаки. Ботовете се стартират на съответните машини и при стартирането си се включват в тестовия канал, през който атакуващият бот ще задава командите. Проведените тестове показват работоспособността на разработената система.

Г.8.14 Ю.Алексиева, Х.Вълчанов. Симулатор на ботнет DoS атаки в мрежова среда. Автоматика и информатика, N:2, 2016, 43-52. ISSN 0861-7562

Статията представя някои аспекти от реализацията на симулационна среда за генериране на IRC DoS мрежови атаки. Архитектурата на системата е разпределена, състояща се от множество ботове, които се изпълняват на отделни възли и си комуникират чрез IRC канали. Всеки бот се състои от следните компоненти:

- Ядро – реализира базовата функционалност на бота, като координира работата на отделните му компоненти.
- Команден интерпретатор – реализира вътрешната система за приемане и обработка на команди от отдалечени потребители.
- Генератори на атаки – реализират трите базови типа атаки- TCP, UDP и ICMP.
- Комуникационна подсистема – реализира комуникацията към другите ботове в мрежата, използвайки механизма на сокетите.
- IDE – графичен интерфейс на бота, позволяващ взаимодействие с потребител на локалната машина.

Системата може да действа с поне един свързан бот към мрежата до неограничен брой ботове. Ботът се стартира локално на всяка машина и се контролира чрез IRC канала, към който се свързва. Този IRC канал представлява C&C центъра на ботнета. Управлението на ботнет DoS атаките става отдалечено чрез подаване на команди в зададения IRC канал. Всички ботове получават една и съща команда и всеки я изпълнява независимо от останалите с цел по-голяма ефективност на атаката. След изпълнение на атаката ботовете използват IRC протокола и се свързват към потребителя на канала, който е задал командата, като му изпращат съобщение с отчет за извършената дейност.

Тестването на симулационната среда е направено по отношение на два основни типа мрежова инфраструктура - равнинна и маршрутизирана. Тестовете са извършени в две насоки: за тестване на комуникацията между ботовете и за реализация на мрежови атаки. Ботовете се стартират на съответните машини и при стартирането си се включват в тестовия канал, през който атакуващият бот ще задава командите. Втората група тестове изследва функционалността и ефикасността на генерираните мрежови атаки. Целта е да се определи ефикасността на симулационната среда за насищане на комуникационния канал с пакети. Атаките са осъществени по отношение на трите базови протокола TCP, UDP и ICMP. Информацията за резултатите от трафика за всяка атака се получава от обобщени отчети на WireShark.

Като насоки за бъдеща работа се предвижда разширяване с генератори на други мрежови атаки, както и добавяне на възможности за реализиране на DDoS атаки. Друга насока е използването на нишки за повишаване ефикасността на симулатора.

Г.8.15 Димитър Н. Тодоров, Христо Г. Вълчанов, Разпределена система за търсене в локална мрежа. Компютърни науки и технологии, ТУ-Варна, 2016, бр.1, с.32-38, ISSN 1312-3335.

Споделянето на ресурси в локална мрежа е един от начините за ефикасно използване на информацията. Системата за търсене на Windows не предоставя възможност за търсене в локална мрежа. Статията представя архитектурата и особеностите на разпределена система за търсене в локална мрежа.
Архитектурата на системата има разпределен характер. Тя се състои от множество модули за търсене, всеки един от които се изпълнява върху отделна машина в локалната мрежа. Модулите функционират независимо като комуникират помежду си, обменяйки информация за процеса на търсене. Всеки модул се състои от две основни системи- система за индексиране и за търсене.
Системата за индексиране осигурява изграждането на индексните файлове. Тя извършва обработката на заявката за индексиране, извличането на текст от документи, изпълнява алгоритъма за опростяване на думите и реализира изграждането на индексите. Изграждането на индексите е реализирано по два начина: в оперативната и в дисковата памет. Процесът на изграждане на индексите в оперативната памет започва с отстраняване на всички пунктуационни символи от поредния документ и като резултат връща изчистен текст. Отделянето на термините (думите) се реализира чрез генериран от Flex Lexical Analyzer код. Подсистемата за индексиране приключва като записва създадените индексни файлове в дисковата памет на машината, освобождава използваната оперативна памет и услугата преминава в режим на очакване на заявки. Стъпките по индексация в дисковата памет са аналогични на индексирането в оперативната памет, но с някои различия. При този случай има твърдо фиксиран буфер, в който се записват формираните списъци. При запълване на буфера, той се съхранява в дисковата памет, след което се изчиства и процесът по индексация продължава, докато не бъдат индексирани всички файлове в желаната директория и поддиректориите й.
Системата обработва получените заявки за търсене, които могат да бъдат както локално подадени, така и получени по мрежата от други машини. Вече филтрираната заявка, се проверява за наличие на логически оператори, термини с маска, фраза, или е само за търсене на един термин. Всеки термин от заявката се проверява в списъка с изградените индекси. В зависимост от начина на изграждане на индексите се прилага различна функционалност. При индексиране в ОП в паметта първо се зарежда дървото. Всеки символ от заявения термин се търси в дървовидната структура. Последният възел от всяка записана дума съдържа указател към списъка с файлови индекси на термина. При индексиране на диска се зарежда картата с индексите. Първо се проверява началната буква на термина. Ако бъде открито съвпадение се вземат началната и крайна позиции, зарежда се буфер със съответните индекси и терминът се търси в него. Буферът може да се презарежда в зависимост от размера на индексите. Системата позволява изпълнение на няколко типа заявки: търсене по дума, по част от дума (търсене с маска), търсене на фраза, сложни логически изрази.

Г.8.16 Райчинов К., Х. Вълчанов. Архитектура на система за предпазване от мрежови атаки. Proc. of UNITECH’16, Gabrovo, 2016, pp.II-316-II-321. ISSN 1313-230X

Настоящият доклад представя архитектурата и базовата функционалност на система за детектиране и предпазване от атаки IDPS, която предлага достъпно, гъвкаво и ефективно решение за изграждане на подобни системи в различни сфери.

Архитектурата на представяната система има модулен характер. Тя се състои от множество модули, всеки притежаващ определена функционалност:

- Операционна система, върху която работи софтуерната реализация.
- Софтуер, извършващ детектирането на атаки.
- Транслатор на събитията.
- Конзола за представяне на резултатите от анализа на мрежовия трафик.

Като операционна система е използвана Debian Linux базираната система – Raspbian, която е една от официално поддържаните според Raspberry Pi фондацията и предоставя възможности, удовлетворяващи изискванията на настоящата разработка. Под операционната система работи Snort, който е основополагащ елемент при мрежовите IDPS и доказан със своята ефективност софтуер. В тази реализация системата за детектиране на атаки взаимодейства с мрежовия трафик чрез прозрачен мост, през който трафикът преминава.

Тестването на представената система е извършено в реална мрежова инфраструктура. Симулирани са атаки с отказ от обслужване (Denial-of-service – DoS). Генерират се голямо количество заявки (от порядъка на стотици хиляди) от фалшиви източници в наводняващ режим (flooding), за максимално кратък период

Системата е базирана на отворен код, позволяващ изключителна гъвкавост при нужда от модификации и по-нататъшни разработки, но също така и предоставящ възможност за проверка на сигурността на отделните програми и гарантиращ качество на кода от високо ниво.

Цел на бъдеща работа е развитие на архитектурата чрез използване на множество Raspberry Pi сензори, разпределени в различни ключови възли на мрежата, които да пренасочват изходните си данни към централизирана база, оптимизирайки по този начин използването на системата за детектиране на атаки.

Г.8.17 Martin Todorov, Hristo Valchanov. System for generating of DoS network attacks. In Proc. of University of Rousse- 2016, volume 55, book 3.3, pp.7-11, 2016. ISSN 1311-3321

Докладът представя особеностите на система за генериране на мрежови DoS атаки, която предоставя възможност за прилагане на няколко от най-често разпространените и използвани атаки по начин, удобен за усвояване на принципите, стоящи зад тях. Системата е изключително удобна за обучение на мрежови администратори, които могат да тестват мрежата и устройствата си за пропуски в софтуера и конфигурирането им, както и за правилно структурирана топология и защита.

Архитектурата на системата за генериране на атаки има модулен характер. Предимството в сравнение с представените подобни средства е постигане на гъвкава функционалност. От една страна се дава възможност за лесно модифициране на действията при генерирането на атаки, а от друга страна функционалността може да бъде разширявана чрез добавяне на нови модули, реализиращи нови атаки.

- Ядрото реализира базовата функционалност като координира работата на останалите компоненти.
- Модулите за атаки реализират атаките по трите основни протоколи – UDP, TCP и ICMP. Обособяването на три базови модула се определя от спецификата на формиране на пакетите за тези базови протоколи. Всеки от модулите включва отделни библиотеки, които се зареждат към ядрото на системата при нейното стартиране.
- Модулът за UDP атаки реализира UDP flood атака, изпращайки множество UDP пакети към определен порт на целевата машина с цел блокиране на услуга и генериране на голям трафик. Модулът TCP реализира SYN flood атака с цел заемане на ресурси от дестинацията. Модулът за ICMP генерира два типа атаки по този протокол – Ping of Death, при която се препълват буферите на приемащата машина и Smurf атака с цел генериране на огромен трафик в определен мрежови сегмент. Модулът за RAW атаки реализира ARP Spoofing при която се подменя MAC адрес в адресната таблица.

Проведени са експерименти с цел изследване функционалността на представената система в реална мрежова среда. Първата група тестове е за UDP атака. Атакува се машина, на която има стартиран DNS сървър. Втората група е SYN flood атака, при която се използва като услуга Apache HTTP сървър. Третата група е ARP Spoofing атака. При нея се изпраща като отговор фалшив MAC адрес на маршрутизатора в сегмента.

Проведените експерименти показват, че предлаганата система има пълна функционалност по отношение на базовите мрежови атаки.

Цел на бъдеща работа е разработване на допълнителни компоненти за други атаки, изискващи по-специални средства за реализация, както и графичен интерфейс за по-удобно манипулиране с параметрите на атаката.

Г.8.18 Raychinov K., H. Valchanov. Intrusion Detection and Preventing System. In Proc. of TechSys’17, 2017, Plovdiv, pp. II 177-II 180. ISSN Online: 2535-0048

Докладът представя архитектурата и функционалността на система за разпознаване и предпазване от атаки. Основната идея на разработената система е нейното лесно интегриране във всяка мрежова инфраструктура. За тази цел е предложено просто и ефективно решение с помощта на едноплатков компютър Raspberry Pi. Характерна черта на тази система е нейната цялост - концентрацията на всички софтуерни компоненти върху мощен хардуер, същевременно с малък размер и ниска цена на притежание. Предлаганата система отчита удобството и функционалността, която предоставя графичната конзола за анализ на откритите заплахи. Така се реализира архитектура с висока ефективност.
Контролът на трафика през устройството се осъществява от Snort. За по-голяма ефективност на Snort, данните от системата се генерират в двоичен файл в unified2 формат. Този файл се обработва от транслатор, който анализира записаните събития и конкретен трафик и ги пренасочва към база данни. Използваният транслатор е Barnayrd2, работещ паралелно със Snort и пренасочващ нови събития от двоичния файл към базата данни в реално време. Като система за управление на бази данни се използва MySQL. Ключови за всеки базиран на сигнатура IDPS са списъци с правила, според които събитията се определят като имащи възможна степен на заплаха или не. Тези подписи се съхраняват от Snort в няколко типа локални файлове.
Тестването на системата е направено чрез симулация на DoS атака. Атаката се осъществява чрез изпращане само на SYN пакети, като атаката е насочена към рутера в мрежата. Генерират се над 65 000 съобщения за необичайно поведение в мрежата. Броят им продължава да расте, защото транслаторът продължава да чете от двоичния файл и да запълва базата данни с нови съобщения, въпреки, че симулираната атака е приключила. Причината за откритата от системата аномалия са данни в SYN пакети, прихванати от Stream препроцесора на Snort.
Резултатите от проведените експерименти показват, че разработената система е напълно функционална и приложима в реални условия.

Г.8.19 Rajchinov K., H.Valchanov. Embedded Network Intrusion Detection and Preventing System. In Proc. of Conf. Automatics and Informatics’17, Sofia, 2017, pp.249-252. ISSN 1313-1850

Системата за разпознаване и детектиране на мрежови атаки (Network Intrusion Detection and Preventing System - NIDPS) е софтуер, който автоматизира процеса на откриване на атаки и възпиране на възможни инциденти. Докладът представя NIDPS система за интегриране във всяка мрежова инфраструктура на базата на едноплатков компютър Raspberry Pi.
С оглед избягване на физическа достъпност до Razpbery Pi (клавиатура или терминална конзола) е приложено инженерно решение с добавяне на трети WiFi мрежови интерфейс. Стандартният модел B на Raspberry Pi rev.2 има само един Ethernet интерфейс, но за реализиране на системата са нужни поне два, а в най-добрия случай и три мрежови комуникационни интерфейса. За целта са добавени допълнителни външни модули на два USB - един USB-Ethernet преходник, с Ethernet и един USB донгъл за безжична WiFi връзка. По този начин се постига предвидената функционалност чрез два интерфейса, през които да преминава мрежовия трафик, проверяван за проблеми и един интерфейс за достъп до системата, с цел нейното управление. Двата Ethernet интерфейса са без зададени IP адреси и са свързани в мост, работещ на каналния слой на OSI модела, който е прозрачен за горните слоеве.
Мрежовата система за детектиране на атаки е разположена в част от мрежата, през която преминава трафикът между вътрешната и външна за организацията мрежи. Сензорът на софтуера за проверка на преминаващия трафик – Snort, трябва да е зад устройствата, комуникиращи си през криптирана връзка (например VPN маршрутизатори), така, че следеният трафик да не е шифрован. Същевременно, трябва да е разположен зад защитна стена, която да предотвратява евентуални опити за атаки.
Разпознаването на атака е базирано на наличие на аномалии. След детектирането на събитията е необходимо тяхното класифициране, от гледна точка на анализа. В кокретните експерименти, те биват определяни като атака с отказ от обслужване заради обема на заявките от многобройни адреси без завършен диалог по протокола TCP. Тази класификация се осъществява през конзолата за управление – Snorby. След процеса на масова класификация на новите събития те са маркирани като атака с отказ от обслужване (Denial of Service).

Г.8.20 Raychinov K., H. Valchanov. Intrusion Detection and Preventing System. //Journal of the technical University of Sofia Plovdiv Branch, Bulgaria, Fundamental science and Applications,7, 2017, vol.23, pp. 97-100, ISSN 1310 - 8271

Статията представя архитектурата и функционалността на система за разпознаване и предпазване от атаки. Основната идея на разработената система е нейното лесно интегриране във всяка мрежова инфраструктура. За тази цел е предложено просто и ефективно решение с помощта на едноплатков компютър Raspberry Pi. Характерна черта на тази система е нейната цялост - концентрацията на всички софтуерни компоненти върху мощен хардуер, същевременно с малък размер и ниска цена на притежание. Предлаганата система отчита удобството и функционалността, която предоставя графичната конзола за анализ на откритите заплахи. Така се реализира архитектура с висока ефективност.
Контролът на трафика през устройството се осъществява от Snort. За по-голяма ефективност на Snort, данните от системата се генерират в двоичен файл в unified2 формат. Този файл се обработва от транслатор, който анализира записаните събития и конкретен трафик и ги пренасочва към база данни. Използваният транслатор е Barnayrd2, работещ паралелно със Snort и пренасочващ нови събития от двоичния файл към базата данни в реално време. Като система за управление на бази данни се използва MySQL. Ключови за всеки базиран на сигнатура IDPS са списъци с правила, според които събитията се определят като имащи възможна степен на заплаха или не. Тези подписи се съхраняват от Snort в няколко типа локални файлове.
Тестването на системата е направено чрез симулация на DoS атака. Атаката се осъществява чрез изпращане само на SYN пакети, като атаката е насочена към рутера в мрежата. Генерират се над 65 000 съобщения за необичайно поведение в мрежата. Броят им продължава да расте, защото транслаторът продължава да чете от двоичния файл и да запълва базата данни с нови съобщения, въпреки, че симулираната атака е приключила. Причината за откритата от системата аномалия са данни в SYN пакети, прихванати от Stream препроцесора на Snort.
Резултатите от проведените експерименти показват, че разработената система е напълно функционална и приложима в реални условия.

Г.8.21 Y.Aleksieva, H.Valchanov. Network Simulator for Botnet DoS Attacks. Information Technologies and Control, N:1, 2017, 33-40, ISSN 1312-2622

Статията представя някои аспекти от реализацията на симулационна среда за генериране на IRC DoS мрежови атаки. Архитектурата на системата е разпределена, състояща се от множество ботове, които се изпълняват на отделни възли и си комуникират чрез IRC канали. Всеки бот се състои от следните компоненти:

- Ядро – реализира базовата функционалност на бота, като координира работата на отделните му компоненти.
- Команден интерпретатор – реализира вътрешната система за приемане и обработка на команди от отдалечени потребители.
- Генератори на атаки – реализират трите базови типа атаки- TCP, UDP и ICMP.
- Комуникационна подсистема – реализира комуникацията към другите ботове в мрежата, използвайки механизма на сокетите.
- IDE – графичен интерфейс на бота, позволяващ взаимодействие с потребител на локалната машина.

Системата може да действа с поне един свързан бот към мрежата до неограничен брой ботове. Ботът се стартира локално на всяка машина и се контролира чрез IRC канала, към който се свързва. Този IRC канал представлява C&C центъра на ботнета. Управлението на ботнет DoS атаките става отдалечено чрез подаване на команди в зададения IRC канал. Всички ботове получават една и съща команда и всеки я изпълнява независимо от останалите с цел по-голяма ефективност на атаката. След изпълнение на атаката ботовете използват IRC протокола и се свързват към потребителя на канала, който е задал командата, като му изпращат съобщение с отчет за извършената дейност.

Тестването на симулационната среда е направено по отношение на два основни типа мрежова инфраструктура - равнинна и маршрутизирана. Тестовете са извършени в две насоки: за тестване на комуникацията между ботовете и за реализация на мрежови атаки. Ботовете се стартират на съответните машини и при стартирането си се включват в тестовия канал, през който атакуващият бот ще задава командите. Втората група тестове изследва функционалността и ефикасността на генерираните мрежови атаки. Целта е да се определи ефикасността на симулационната среда за насищане на комуникационния канал с пакети. Атаките са осъществени по отношение на трите базови протокола TCP, UDP и ICMP. Информацията за резултатите от трафика за всяка атака се получава от обобщени отчети на WireShark.

Като насоки за бъдеща работа се предвижда разширяване с генератори на други мрежови атаки, както и добавяне на възможности за реализиране на DDoS атаки. Друга насока е използването на нишки за повишаване ефикасността на симулатора.

Г.8.22 Valchanov H., D.Trifonov. Performance analysis of Virtualization and Containerization Platforms for Big Data Processing. Proc. of UNITECH’17, Gabrovo, 2017, pp. 203-208, ISSN 1313-230X

Докладът представя сравнителен анализ на производителността на системи за обработка на големи обеми данни, изградени на базата на виртуални машини и на контейнери.
Разликата между виртуализация и контейнеризация е основно в мястото на виртуализационния слой и начинът, по който системните ресурси се използват. Контейнеризацията, наричана още „контейнерно базирана виртуализация”, “паравиртуализация” или “виртуализация на приложения”, представлява виртуализационен метод за внедряване и изпълнение на разпределени приложения на нивото на операционната система, без необходимостта от стартиране на цяла виртуална машина за всяко приложение. Вместо това, множество изолирани системи, наречени контейнери, са изпълнявани на един единствен хост и достъпват ядрото на операционната система.
Като платформа за виртуализация е избран хипервайзорът VMware ESXi 6.5, а за контейнеризация – Docker. Изборът на тези две платформи е продиктуван от тяхното широко използване, високата им производителност и възможности.
Тестовете са така подбрани, че да покрият изискванията към различните хардуерни устройства: процесор, памет и устройства за съхранение. Първият тест тества производителността на системата при обработка на данни в полуструктуриран вид. Такава обработка се налага постоянно в Big data системите, понеже входните данни доста често идват от разнородни източници и са в различни формати. Резултатите показват предимство от ~4,5% за контейнеризацията и Docker. Вторият тест е комплексен по отношение на процесорна обработка и входно/изходни операции – Hive dataset import. Този тест използва програмния модел MapReduce. Прочита се голям по размер файл и се импортира в таблица на нерелационна база данни MongoDB. Следва да се отбележи, че този тест не дава ясен резултат за предимство на виртуализацията или контейнеризацията. Протича по-бързо от първия тест, понеже в базата от данни текстът се съхранява в различен формат, а и обработката на данните се свежда единствено до извличане от текста и запис в базата. Липсват фазите на сортиране и комбиниране на сортираните резултати.
Получените резултати дават основание да се твърди, че контейнеризацията дава малко по-добри резултати в повечето случаи.

Г.8.23 Valchanov H., D.Trifonov. Performance analysis of Virtualization and Containerization Platforms for Big Data Processing. Proc. of UNITECH’17, Gabrovo, 2017, Selected papers, pp. II283-II288, ISSN 2603-378X.

Докладът представя сравнителен анализ на производителността на системи за обработка на големи обеми данни, изградени на базата на виртуални машини и на контейнери.
Разликата между виртуализация и контейнеризация е основно в мястото на виртуализационния слой и начинът, по който системните ресурси се използват. Контейнеризацията, наричана още „контейнерно базирана виртуализация”, “паравиртуализация” или “виртуализация на приложения”, представлява виртуализационен метод за внедряване и изпълнение на разпределени приложения на нивото на операционната система, без необходимостта от стартиране на цяла виртуална машина за всяко приложение. Вместо това, множество изолирани системи, наречени контейнери, са изпълнявани на един единствен хост и достъпват ядрото на операционната система.
Като платформа за виртуализация е избран хипервайзорът VMware ESXi 6.5, а за контейнеризация – Docker. Изборът на тези две платформи е продиктуван от тяхното широко използване, високата им производителност и възможности.
Тестовете са така подбрани, че да покрият изискванията към различните хардуерни устройства: процесор, памет и устройства за съхранение. Първият тест тества производителността на системата при обработка на данни в полуструктуриран вид. Такава обработка се налага постоянно в Big data системите, понеже входните данни доста често идват от разнородни източници и са в различни формати. Резултатите показват предимство от ~4,5% за контейнеризацията и Docker. Вторият тест е комплексен по отношение на процесорна обработка и входно/изходни операции – Hive dataset import. Този тест използва програмния модел MapReduce. Прочита се голям по размер файл и се импортира в таблица на нерелационна база данни MongoDB. Следва да се отбележи, че този тест не дава ясен резултат за предимство на виртуализацията или контейнеризацията. Протича по-бързо от първия тест, понеже в базата от данни текстът се съхранява в различен формат, а и обработката на данните се свежда единствено до извличане от текста и запис в базата. Липсват фазите на сортиране и комбиниране на сортираните резултати.
Получените резултати дават основание да се твърди, че контейнеризацията дава малко по-добри резултати в повечето случаи.

Г.8.24 H.Valchanov. Power Management of a Virtual Infrastructure. Компютърни науки и технологии, v.1, 2018 77-82, ISSN 1312-3335

Докладът представя архитектурата на система за управление на захранването и виртуалните машини в инфраструктура, изградена на базата на VMware ESXi клъстер. Представена виртуалната инфраструктура, която е необходимо да бъде управлявана. Тази инфраструктура се използва в учебния процес на студентите от специалности „Компютърни системи и технологии” и „ Софтуерни и интернет технологии” при ТУ-Варна. Основният компонент е клъстер от високопроизводителни сървъри HP BL460c Gen8 с VMware виртуализационна платформа, базирана на ESXi 6.0. Върху тях се изпълняват над 50 виртуални машини със специфично предназначение за учебния процес. Виртуалните машини са под Windows и Linux ОС. В комуникационния шкаф са монтирани дисков масив, и две сървърни машини. Захранването се поддържа от непрекъсваемо устройство Eaton 9SX5000.

Архитектурата на системата за управление на захранването се състои от няколко модули. Модулите са реализирани като сървиси и скриптове, които се изпълняват на отделен сървър под управлението на Windows Server 2012 R2. Модулът SNMP trap receiver се изпълнява като услуга във фонов режим. Неговото предназначение е да следи в мрежата за наличието на SNMP съобщение с определен код. Това съобщение се генерира от UPS. UPS е конфигуриран да премине в режим на изключване при определен оставащ капацитет на батерията (за целите на изследването това е 65%).

Ако са налице всичките условия, то трябва да се премине към действията по нормалното изключване на инфраструктурата. Стартира се специален скрипт, който изпълнява необходимата последователност от действия. Скриптът е реализиран с използване на PowerShell под Windows и пакета на VMware – VMwareCLI. Първоначално се преминава през фазата на автентикация, осъществена от модула Authentication. Неговото предназначение е да осигури коректното идентифициране и логване към различните системи. След успешното свързване се преминава към същинската последователност на изключване. Тя се реализира от Shutdown модула, който се изпълнява като скрипт.

При всяко действие се записва информация в журнален файл, която може да бъде използване впоследствие от администратора за анализ.

Функционалността на представената система й не зависи от вида на използваното непрекъсваемо захранване. Системата е модулна, което позволява лесно нейно бъдещо разширяване. Системата е внедрена и в продължение на 1 година е доказала своята работоспособност.

Като насоки за бъдеща работа се предвижда разработване на графичен потребителски интерфейс, както и интегрирането й като сървис в операционната система ESXi.

Г.8.25 Todorov D., H.Valchanov. Routing and Traffic Load Balancing in SDN-NFV Networks. Proc of International Conference Applied Computer Technologies, 2018, pp.127-130. ISBN 978-608-66225-0-3

Докладът представя обзор на методи за маршрутизиране и балансиране на натоварването на трафика в SDN-NFV мрежи.
Представено е широкото използване на виртуализационните технологии за изграждане на виртуални мрежи и необходимостта от средства и методи за автоматизиране на тяхното управление.
Дискутирани са архитектурите на софтуерно дефинираните мрежи (SDN) и виртуализацията на мрежовите функции (NFV). Описани са функционирането на SND контролерите, взаимодействието с управляваните устройства, както и приложението на стандартизирания интерфейс между контролера и устройствата, базиран на протокола OpenFlow.
Разгледани са характерните особености на SDN техники за маршрутизация, като Virtual Routers as a Service (VRS), Routing as a Service (RaaS), RouteFlow Routing Control Platform through SDN (RFCP), SoftRouter, RouteFlow IP routing.
Един от основните проблеми на компютърните мрежи е балансирането на натоварването на трафика. В доклада са разгледани техники за балансирано натоварване по отношение на две нива: NFV и SDN.
По отношение на NFV са представени решения чрез избор на пътища за маршрутизиране, на разпределяне на трафика между NFV системи и чрез използване на алгоритъма ORBIT.
По отношение на балансираното натоварване при SDN са представени решения като най-слабо натоварения сървър в реално време (RLS), използване на контролер за анализ на отговорите от OpenFlow комутаторите, прилагането на Dynamic Load Balancing algorithm, използване на евристични методи.

Г.8.26 Trifonov D., H. Valchanov. Virtualization and Containerization Systems for BigData. In Proc. of TechSys'18, 2018, Plovdiv, pp.II157 – II160, ISSN Online: 2535-0048

Докладът акцентира върху приложението на виртуализационни и контейнеризационни системи за големи обеми от данни. Представено е изследване на производителността на голяма система за обработка на данни, внедрена върху виртуални машини и контейнери.
Системата за обработка на данни е Hadoop. Hadoop е софтуер с отворен код, разработен в Java. Hadoop разполага с редица инструменти за изпълнение на код и скриптове на различни езици за програмиране. Състои се от две основни части - част за съхранение и част за обработка на данни. Компонентът за съхранение е разпределената файлова система - Hadoop Distributed File System (HDFS). Частта за обработка на данни е MapReduce. Това е програмен модел за паралелна обработка на множество задачи. Един от основните аспекти на програмирането на MapReduce е, че MapReduce разделя задачите по такъв начин, че те позволяват паралелното им изпълнение в разпределена система от изчислителни възли. Противно на традиционните системи за управление на релационни бази данни, които не могат да се разширяват, за да обработват големи количества данни, програмирането в средата на Hadoop MapReduce позволява на потребителите да изпълняват приложения на огромен брой машини, което включва и обработката на хиляди терабайти данни.
Представено е експериментално изследване на производителността на Hadoop върху хипервайзор VMware ESXi и платформа Docker. Проведени са три вида тестове. Първият тест изследва производителността при обработка на неструктурирани данни. Тестът е Java приложение (WordCount), изпълнявано директно от Hadoop , като се извиква MapReduce. Данните за обработка са едни от най-новите архиви на Wikipedia (EN), като се използва само английско съдържание. Предоставя се като bz2 архив, в който има xml файл от приблизително 62 GB. Резултатите показват предимство от ~4,5% за контейнеризация и Docker. Това се очаква, като се има предвид, че по време на теста се използва максимално количество памет и всяка виртуална машина използва приблизително 1,3 GB памет, която иначе се използва от Hadoop. Достъпът до дисковото хранилище за виртуализация е потенциално малко по-бавен, което също леко увеличава преднината на Docker.
Вторият тест е сложен по отношение на работата на процесора и I/O операциите – импортиране на набор от данни на Hive. Този тест отново използва MapReduce. Той чете съдържанието на текстовия файл enwiki-20170701-pages-articles-multistream.xml и го импортира в таблица без релации MongoDB. Третият тест оценява системата за съхранение. Изпълнява се в две части: TestDFSIO-write записва 100GB данни във файловата система Hadoop-HDFS, а TestDFSIO-read ги чете обратно. Леко предимство се отчита отново за контейнеризацията поради факта, че достъпът до диска през виртуалния дисков контролер е малко по-бавен, отколкото през интерфейса, предоставен от контейнеризацията.

Г.8.27 Y.Aleksieva, H.Valchanov. BOTNET DETECTION SYSTEM BASED ON GENETIC ALGORITHMS. In Proc. of Conf. Automatics and Informatics’18, Sofia, 2018, pp.129-139. ISSN 1313-1850.

Докладът представя някои аспекти на реализацията на хост-базирана система за откриване на Ботнет атаки. Системата използва техника за откриване на аномалии в поведението на базата на вариация на генетичен алгоритъм. Архитектурата на представяната система има модулен характер:

- Ядро – реализира базовата функционалност на системата, като координира работата на отделните компоненти.
- Команден интерпретатор – реализира вътрешната система за приемане и обработка на командите, зададени от потребителя.
- Комуникационна подсистема – реализира комуникацията между потребителя и администратора / между системата и администратора.
- GUI – графичен интерфейс на системата, позволяващ взаимодействие с потребителя на локалната машина.

Предложен е алгоритъм за откриване на Ботнет атаки извършва детектиране на spoofing атака чрез специфична вариация на генетичен алгоритъм, като представената вариация се базира на генетичния оператор селекция, оценявайки всички индивиди в поредните поколения на база на аналитично определена фитнес функция. Алгоритъмът създава хромозоми от получените пакети, изследва ги внимателно и така засича промени във фенотипа и мутация. В случая на използвания генетичен алгоритъм, системата третира получените пакети като фенотип. Алгоритъмът извлича от получения пакет (фенотипа) неговите гени и комбинира тези гени във верификационна хромозома. Когато се извършва засичане на spoofing атака, алгоритъмът изследва хедърите на всеки пакет и извлича атрибути (гени), като IP адрес, MAC, TTL, Protocol number и др.

Фитнес функцията на алгоритъма представлява аналитично изчисление между две хромозоми, получени от един и същ адрес. Двете хромозоми биват двоично сравнени бит по бит, изчислявайки се математически фитнес нивото на получения пакет. Когато системата получи пакет за първи път от даден IP адрес, гените му формират хромозома и тя се запазва със 100% фитнес ниво (ниво на годност). Хромозомите на следващите пакети, получени от същия източник, се сравняват с хромозомата на този пакет в базата данни, прилагайки фитнес функцията и аналогично се изчислява фитнес нивото на пакета. Ако фитнес нивото на новия приет пакет е по-малко от това на пакета, получен по-рано, то следователно има налице външна намеса, при което се генерира аларма.

Тестването на представената система е извършено в реална мрежова инфраструктура. Резултатите от тестването показват, че алгоритъмът функционира коректно, но конкретната мрежа трябва да бъде внимателно анализирана, за да се избере подходящо фитнес ниво. В атакувана среда хост-базираната система открива веднага атаката и сигнализира адекватно.

Цел на бъдеща работа е разширение на функционалността на системата, предоставяща още възможни техники за откриване на аномалии, като добавяне на анализ на интегритета на данните.

Г.8.28 Aleksieva Y., H.Valchanov. Anomaly Based Botnet Setection System. Proc. of UNITECH'18, Gabrovo, 2018, vol.2, pp. II123-II127, ISSN 1313-230X

Докладът представя някои аспекти на реализацията на хост-базирана система за откриване на Ботнет атаки. Системата използва техника за откриване на аномалии в поведението на базата на вариация на генетичен алгоритъм.
Алгоритъмът за откриване на атаки извършва детектиране на spoofing атака чрез специфична вариация на генетичен алгоритъм, като представената вариация се базира на генетичния оператор селекция, оценявайки всички индивиди в поредните поколения на база на аналитично определена фитнес функция. Алгоритъмът създава хромозоми от получените пакети, изследва ги внимателно и така засича промени във фенотипа и мутация. В случая на използвания генетичен алгоритъм, системата третира получените пакети като фенотип. Алгоритъмът извлича от получения пакет (фенотипа) неговите гени и комбинира тези гени във верификационна хромозома. Когато се извършва засичане на spoofing атака, алгоритъмът изследва хедърите на всеки пакет и извлича атрибути (гени), като IP адрес, MAC, TTL, Protocol number и др.
Фитнес функцията на алгоритъма представлява аналитично изчисление между две хромозоми, получени от един и същ адрес. Двете хромозоми биват двоично сравнени бит по бит, изчислявайки се математически фитнес нивото на получения пакет. Когато системата получи пакет за първи път от даден IP адрес, гените му формират хромозома и тя се запазва със 100% фитнес ниво (ниво на годност). Хромозомите на следващите пакети, получени от същия източник, се сравняват с хромозомата на този пакет в базата данни, прилагайки фитнес функцията и аналогично се изчислява фитнес нивото на пакета. Ако фитнес нивото на новия приет пакет е по-малко от това на пакета, получен по-рано, то следователно има налице външна намеса, при което се генерира аларма.
След като хромозомата е генерирана, трябва да се изчисли нейното фитнес ниво. Нека, например, хромозомата на получен пакет от източник за първи път, която е генерирана от алгоритъма, е 1010101010. След това, ако се получи друг пакет от източник със същия IP адрес, се очаква неговата хромозома да е почти същата, ако ли не – то поне близка до предишната. Това се дължи на факта, че MAC адресите на източника и дестинацията трябва да са еднакви, а TTL и HopCount - приблизително еднакви. Също така, идентификаторът на пакета се очаква да е по-голям от идентификатора на предишния пакет. Ако всичко това е налице, очаква се вторият получен пакет да има почти същата хромозома. Например, нека хромозомата на втория получен пакет е 1100101001. От сравнението се виждат 4 разлики в хромозомите. В този случай вторият пакет притежава фитнес ниво 60%, което означава, че е бил значително променен и системата ще генерира аларма.
Тестването на представената система е извършено в реална мрежова инфраструктура. Резултатите от тестването показват, че алгоритъмът функционира коректно, но конкретната мрежа трябва да бъде внимателно анализирана, за да се избере подходящо фитнес ниво.

Г.8.29 Trifonov D., H.Valchanov. VIRTUALIZATION AND CONTAINERIZATION SYSTEMS FOR BIG DATA. //Journal of the technical University of Sofia Plovdiv Branch, Bulgaria, Fundamental science and Applications,7, 2018, vol.24, pp. 129-132. ISSN 1310 – 8271.

Статията акцентира върху приложението на виртуализационни и контейнеризационни системи за големи обеми от данни. Представено е изследване на производителността на голяма система за обработка на данни, внедрена върху виртуални машини и контейнери.
Системата за обработка на данни е Hadoop. Hadoop е софтуер с отворен код, разработен в Java. Hadoop разполага с редица инструменти за изпълнение на код и скриптове на различни езици за програмиране. Състои се от две основни части - част за съхранение и част за обработка на данни. Компонентът за съхранение е разпределената файлова система - Hadoop Distributed File System (HDFS). Частта за обработка на данни е MapReduce. Това е програмен модел за паралелна обработка на множество задачи. Един от основните аспекти на програмирането на MapReduce е, че MapReduce разделя задачите по такъв начин, че те позволяват паралелното им изпълнение в разпределена система от изчислителни възли. Противно на традиционните системи за управление на релационни бази данни, които не могат да се разширяват, за да обработват големи количества данни, програмирането в средата на Hadoop MapReduce позволява на потребителите да изпълняват приложения на огромен брой машини, което включва и обработката на хиляди терабайти данни.
Представено е експериментално изследване на производителността на Hadoop върху хипервайзор VMware ESXi и платформа Docker. Проведени са три вида тестове. Първият тест изследва производителността при обработка на неструктурирани данни. Тестът е Java приложение (WordCount), изпълнявано директно от Hadoop , като се извиква MapReduce. Данните за обработка са едни от най-новите архиви на Wikipedia (EN), като се използва само английско съдържание. Предоставя се като bz2 архив, в който има xml файл от приблизително 62 GB. Резултатите показват предимство от ~4,5% за контейнеризация и Docker. Това се очаква, като се има предвид, че по време на теста се използва максимално количество памет и всяка виртуална машина използва приблизително 1,3 GB памет, която иначе се използва от Hadoop. Достъпът до дисковото хранилище за виртуализация е потенциално малко по-бавен, което също леко увеличава преднината на Docker.
Вторият тест е сложен по отношение на работата на процесора и I/O операциите – импортиране на набор от данни на Hive. Този тест отново използва MapReduce. Той чете съдържанието на текстовия файл enwiki-20170701-pages-articles-multistream.xml и го импортира в таблица без релации MongoDB. Третият тест оценява системата за съхранение. Изпълнява се в две части: TestDFSIO-write записва 100GB данни във файловата система Hadoop-HDFS, а TestDFSIO-read ги чете обратно. Леко предимство се отчита отново за контейнеризацията поради факта, че достъпът до диска през виртуалния дисков контролер е малко по-бавен, отколкото през интерфейса, предоставен от контейнеризацията.

Г.8.30 Hristo Valchanov, Veneta Aleksieva, Jan Edikyan, Study of wireless networks security, UNITECH’19, 15-16 November 2019, GABROVO, vol.II, pp. 7-11, ISSN 1313-230X

Безжичната мрежа позволява лесно изграждане на домашни мрежи и мрежи в малки предприятия, базирани на стандарта IEEE 802.11. Въпреки това, безжичните мрежи са лесно податливи на атаки срещу тяхната сигурност. Това изисква анализ на проблемите и създаване на препоръки за подобряване на тяхната сигурност. Тази статия представя методология и изследване на сигурността на безжичната мрежа във Варна. Информацията е събрана с помощта на техниката war-driving. Получените резултати се анализират и сравняват с тези от предишни проучвания.

Системата за събиране на данни е изградена на базата на едноплатков компютър Raspberry Pi 3 Model B, с процесор ARM Cortex-A53, 1.2GHz, вградена Wi-Fi и Bluetooth функционалност. За целта на реализацията е нужно записването на позицията на всяка безжична точка за достъп. Избраният GPS модул, поради поддръжка на стандарта NMEA 0183, дълготрайна батерия и голяма памет е Holux M-1200E. За сканиране на безжичните мрежи се използва модул CanaKit WiFi Module. За да се осигури продължително захранване на едноплатковия компютър, се използва портативна батерия Canyon CNS-TPBP5DG с капацитет 5000mAh.

Сканирането на безжичните мрежи е реализирано посредством софтуер с отворен код Kismet. Софтуерът е компилиран и инсталиран под операционната система Raspbian OS. Получените от Kismet данни се записват в netxml формат. Събраната информация са конвертира чрез скрипт на Python в csv формат. Това е необходимо, за да могат данните да се представят в табличен вид с цел по-лесна обработка и анализ чрез Microsoft Excel. Избраният район за анализ включва централната част на гр. Варна, тъй като в нея се намират по-голяма част от офисите и голяма част от живущите. Също така, районът съвпада с проведено подобно изследване от 2008г., с цел сравнение на получените резултати.

Резултатите показват значително увеличаване на сигурността на WiFi мрежите в града, но въпреки това има какво още да се подобри в тази насока.

Причините за подобряване на сигурността могат да се разгледат в две насоки. Първо, производителите предлагат устройства, които по подразбиране имат конфигуриран протокол WPA2. Второ, по-големите организации имат ИТ отдели, които се грижат за сигурността. Въз основа на резултатите за откритите SSID, смесен режим WPA/WPA2 и WPS, може да се заключи, че повечето от анализираните WiFi мрежи принадлежат на обикновени потребители, които нямат достатъчно знания за сигурност.

Основните препоръки могат да бъдат представени в следните направления:

1. Да се използва само метод WPA2 за криптиране.
2. Да се деактивира WPS за всички устройства.
3. Да се избира сложна парола.
4. Да се актуализира софтуера на устройствата до последната версия.
5. Да се информират потребителите за проблемите в сигурността на Wi-Fi мрежата.

Г.8.31 Veneta Aleksieva, Hristo Valchanov, Yuri Dimitrov, Study of smart watch interfaces, UNITECH'19,15-16 November 2019, GABROVO, vol.II, pp. 12-16, ISSN 1313-230X

Умните часовници са носими устройства с малки размери. Техният размер на дисплея и ограниченото пространство за контроли за въвеждане изискват специално внимание към процесите на разработване на интерфейси на устройството. Изследването в този доклад има за цел да сравни два различни подхода в дизайна на интерфейса - нов интерфейс, базиран на активиране и управление с два пръста на сензорен интерфейс на безела на устройството, чувствителен на допир (повърхността, която е около дисплея), със стандартен интерфейс за въвеждане в стил „ръчен часовник" въз основа на странични бутони. Целта на това проучване е да се сравни процеса на взаимодействие човек– смарт часовник, когато един и същ потребител изпълнява една и съща задача при използване на два прототипа на смарт часовници, разработени за целта.

За целите на изследването са изработени два експериментални прототипа с различни входни интерфейси - „Нов" с интерфейс с чувствителен на допир безел, който е разработен за предишно изследване, и „Стандартен" с четири бутона отстрани на устройството, който е проектиран и разработен за настоящото изследване. За да са напълно сравними резултатите от изследванията, двата модела са изработени въз основа на един и същ 3D модел на ръчен часовник. Експерименталните модели се управляват от компютър Arduino Mega 2560. Разработен е софтуер за управление на всеки един модел на база на Arduino. И двата софтуерни продукта разпознават четири основни командни интерфейса за взаимодействие - „Нагоре", „Надолу", „Избери" и „Назад".

Тестовата група се състои от 10 доброволци, ползващи дясна ръка за работа с прототипите. Средната възраст на участниците в експериментите е 37,6 години. Всички експерименти са проведени при равни други условия - в едно и също помещение, без наличие на изкуствено осветление.

Изследването на разработените модели на интерфейси на смарт часовници се проведе в три етапа – на първия етап се сравняват времето и точността на изпълнение на проста задача (избор само на една функция), на втория етап се сравняват времето и точността на изпълнение на сложна задача (избор на функция в няколко стъпки), а на третия етап доброволците дават субективна оценка за комфорта на работа с двата интерфейса.

Изводите от проучването могат да се обобщят в следните:

• Новият експериментален модел превъзхожда стандартния по скорост на работа. Когато наборът от команди е по-дълъг, ползата от използването на новия модел на интерфейс е по-голяма.

• Коефициентите на грешки при работа с новия модел са по-високи от тези при използване на стандартния модел. Причината може да е фактът, че традиционните интерфейси със странични бутони на електронен часовник са познати на повечето хора, но интерфейсът на новия модел с чувствителен на допир пръстен е нещо ново за тях.

• Оценката на потребителите за комфорта на работа с двата интерфейса е по-висока за работата със стандартен модел.

Г.8.32 В. Алексиева, Х.Вълчанов, А. Хулиян, Приложение на интелигентни договори базирани на Ethereum блокчейн за целите на застрахователни услуги, // Информатика и иновативни технологии, сс.7-14 бр.1(1),2019, ISSN 2682-9517

Настоящата статия представя експериментална реализация на интелигентни договори за застрахователни услуги върху Ethereum блокчейн. Авторите представят класически модел на застрахователна услуга и изтъкват неговите недостатъци. На тази основа предлагат модел за застрахователни услуги, базиран на блокчейн технологии. Представена е експериментална реализация върху Ethereum блокчейн.

Процесът на обработка на иск може да бъде подобрен, използвайки интелигентни договори и блокчейн технология. Информацията за настъпила щета може да се изпраща от застрахования или директно от сензори, монтирани в застрахования обект (smart asset), към автоматизирано приложение за обработка на иск. За съответните застрахователни политики, които се осигуряват от интелигентния договор (smart contract), клиентът ще получи обратно потвърждение в реално време. Искът се обработва автоматично от smart contract на базата на зададена бизнес логика, като се използва предоставена от застрахования информация. DLT автоматично използва допълнителните източници (статистики, отчети) за оценка на иска и изчисляване на щетата. В зависимост от застрахователната политика, smart contract може автоматично да изчисли персоналната отговорност. При известни ситуации smart contract може да активира допълнителна оценка на иска. Ако искът е одобрен, плащането към застраховани се инициира чрез smart contract.

Предимствата на новия подход, базиран на интелигентни договори върху блокчейн технология, могат да се разгледат в няколко аспекта. Изпращането на иска е опростено и автоматизирано. Благодарение на директния обмен на информация за щета между застрахователите, DLT премахва необходимостта от участие на брокери и редуцира времето за обработка на иска. Вградената бизнес логика в интелигентния договор в блокчейна елиминира необходимостта вещите лица да преглеждат всеки иск (с изключение на специфични ситуации). Застрахователят има достъп до историята за произхода на щетите, което помага за идентифициране на потенциални опити за измама. Използваната информация е интегрирана, благодарение на възможностите на DLT да обединява данни от множество доверени източници. Процесът на плащане на щетата е автоматизиран от интелигентния договор върху блокчейна, без необходимостта от използване на посредник.

Изложени са предимствата и недостатъците на използване на частен и публичен блокчейн, както и на комбинирани решения с 2 блокчейна (за автоматизиране на бек-офис операциите да се ползва частен блокчейн, а за управление на автоматичните плащания със съществуващи криптовалути или когато има нужда да се осигури доверие да се ползва публичен блокчейн).

Представеното решение е с публичен блокчейн Ethereum.

Г.8.33 V. Aleksieva, A. Huliyan, H. Valchanov, An approach of Crypto-token for Smart Contract based on Ethereum Blockchain, Journal of the Technical University – Sofia, Plovdiv branch, Bulgaria, "Fundamental Sciences and Applications", Vol 25 No 1 (2019), pp.1-7, ISSN 2603-459X, https://journals.tu-plovdiv.bg/index.php/journal

Предложената статия представя решение за създаване на децентрализиран токен за прилагане на интелигентен договор, базиран на блокчейн Ethereum. Създаден е уеб базиран интерфейс за първоначално предлагане на монети (ICO). В експериментална среда бяха проведени изследвания за различни сценарии. Представени са резултатите.
Този интелигентен договор и уеб-базиран интерфейс са представени в Г8.17 и Г8.18. В тази статия са представени експериментални тестове и резултати за неговата функционалност.
Първата част от тестовете е свързана с правилната работа на интелигентния договор - баланс на сметката, прехвърляне на токени и т.н. Има няколко инструмента за автоматизирани интелигентни договори (написани на Solidity) за тестване на уязвимости на сигурността на базата на анализ на ниво код.
В подхода на Reza е дадено обобщение на четирите най-подходящи инструмента, които е възможно да се използват в експериментите, а именно Oyente, Mythril, Securify и SmartCheck. Въпреки това, степента на строгост на оценката, вариращa от синтактична, евристична, аналитична до напълно официална, се отнася до основната техника за тестване на сигурността на дадения инструмент и до този момент изследователите се доверяват на внедрените в инструменти за тестване на стабилността. Truffle (и Solidity) има вграден механизъм за тестване на интелигентни договори, написан на JavaScript, който тук се използва.
За директно тестване на трансфер, 250 000 токена се прехвърлят от адреса на администратора до адреса на получателя. След като прехвърлянето е извършено, събитието се улавя и проверява за тип „Transfer". Ако този тест е успешен, балансът на адреса на получателя се проверява за наличие на прехвърлени токени. Делегираната проверка за прехвърляне е подобна на проверката за директно прехвърляне. Първо, 100 токена се прехвърлят от адреса на администратора на адреса, от който ще бъде разрешен делегиран трансфер – адрес_1. Позволено е да се изразходват 10 символа от адрес_3, който ги изпраща на адрес_4. След извършване на тези действия, той се очаква адрес_1 да има 90 токена, адрес_2 - 0, а адрес_3 да е 10 токена. Показани са резултатите от тяхното изпълнение с грешни и правилни параметри.
В истинския блок на Ethereum е фиксирано времето за в блок, но има динамична промяна на трудността в зависимост от това колко енергия е включена в мрежата. Тестовете бяха проведени в локална мрежа с flat топология. Клиентът се свързва със сървъра Metamask. Параметрите на компютрите са Apple Mac Book Pro Late 2011 Specs, Core i5 (I5-2435M) 2.4GHz 2/4 ядра/нишки, 4GB DDR3 1333Mhz RAM. В Метамаск при изпращане на етер за закупуване на токен се използва протокол TLSv1.2. В статията е представена мрежовата комуникация между клиент и Metamask сървър по време на успешна транзакция на токени.

Г.8.34 D.Dinev, V.Aleksieva, H.Valchanov, K.Genov, LoRaWan Network Mobility Software Simulation Tool, // Computer Science and Tecjnologies, Bulgaria N.1,2021, pp.31-38, ISSN 1312-3335

В този доклад е представен симулатор за реализиране на мобилност в LoRaWan мрежи. Някои от многото изисквания на концепцията за IoT са изпълнени в широколентови мрежи с ниска мощност (LPWAN) - ниска цена, енергийна ефективност и голямо покритие на зоната. Едно от най-проучените внедрявания на LPWAN технологиите са широкообхватните мрежи с голям обхват (LoRaWan). LoRaWan е сравнително нова технология с много предимства и недостатъци. Някои недостатъци могат да бъдат отстранени чрез изучаване на технологичните граници и създаване на симулатори, чрез които да се продължи по-нататъшното развитие на технологиите. В тази статия е предложен симулатор за внедряване на предаването в LoRaWan, който реализира мобилност на крайните устройства и намиране на най-добрия маршрут между крайните устройства преди и след това. Внедрените алгоритми за намиране на най-добрия маршрут между крайни устройства и симулиране на мобилност се използват за изучаване и подобряване на QoS параметрите в LoRaWan мрежи.

Мобилността (хендовер) в безжичните мрежи може да бъде причинена от много причини - физическо движение на свързаното устройство, промяна на характеристиките на мрежата и т.н. Има два различни типа хендовер в зависимост от вида на мрежата за достъп, която принадлежи на всеки PoA - хоризонтален и вертикален хендовер.

Симулаторът изпълнява мобилността в мрежата LoRaWan въз основа на предложения алгоритъм за мобилност. Качеството на сигнала се проверява за реализиране на мобилност. Стартирането на процедурата за хендовер се основава на стойността на силата на получения сигнал (RSS). Хендоверът се осигурява, когато бъде намерен друг gateway (терминал) с по-висока RSS стойност. Мобилността се осъществява на три етапа, съчетавайки предаване на слой 2 и слой 3. Симулационната среда има 6 блока: GUI (Графичен потребителски интерфейс) - включва лесен за използване потребителски интерфейс за симулации;Ядро - основен блок, изпълняващ всички операции на симулатора; Създаване на топология - блок за създаване на топология; Модификация на топология - блок за промяна на съществуващи топологии; Намиране на най-добрия път между крайните устройства - с помощта на „Depth First Search" и „Hassle Free Route Algorithms" може да се намери най-добрият маршрут между крайните устройства;

- Хендовер - с помощта на този блок мобилните крайни устройства се прехвърлят към новите терминални устройства, които отговарят на тези нужди във всяка тяхна посока на движение.

Бутонът „Char" в раздела „handover" отваря прозореца, в който могат да се анализират резултатите от извършените симулации. Представена е диаграма, показваща информация за крайните устройства - техния тип (мобилен, статичен) и състояние (активиран, деактивиран) за всеки терминал, също и диаграма, визуализираща броя на мобилностите, извършени между терминалните шлюзове. Представена е диаграма за MSN, обобщаваща и представяща броя на движенията за всяко мобилно крайно устройство, което е извършило мобилност и др.