

ПРИЛОЖЕНИЯ ЗА МРЕЖОВ МОНИТОРИНГ И ОТСТРАНЯВАНЕ НА ПРОБЛЕМИ В LINUX

Георги Петров, Иван Богомилов

TOOLS FOR NETWORK MONITORING AND NETWORK DEBUGGING IN LINUX

Georgi Petrov, Ivan Bogomilov

Резюме: Целта на тази публикация е да представи обзор върху методите за мрежовият мониторинг и отстраняването на проблеми в IP мрежите използвани от мрежовия архитект и инженер. Познаването на някои основни програмни средства за тестване и наблюдение на трафика в IP мрежите в ОС Linux е от значение при изграждането, разрастването и реконфигурирането на мрежовата инфраструктура. Представени са интегрирани готови функционални инструменти в административни скриптове е основа за създаването на повечето ефективни и мощни инструменти за проверка и отстраняване на неизправности в мрежовата инфраструктура.

Ключови думи: IP мониторинг на трафика, мрежов мониторинг на трафика, откриване и отстраняване на проблеми в мрежите.

Abstract: This publication provides an overview of network monitoring and troubleshooting methods in IP networks used by network architects and engineers. Knowing some basic tools for testing and monitoring traffic in IP networks in Linux is important in building, expanding and reconfiguring network infrastructure. Integrated, ready-made functional tools in administrative scripts are the basis for creating most effective and powerful tools for checking and troubleshooting network infrastructure.

Keywords: IP traffic monitoring, network troubleshooting and debugging.

I. ВЪВЕДЕНИЕ

Изграждането на съвременни високоскоростни и силно разпределени IP (Internet Protocol) базирани мрежи изисква екстензивното използване на разнообразни програмни инструменти за трафичен мониторинг и средства за откриване и отстраняване на проблеми по мрежовото трасе и комутационната инфраструктура. Наред с чисто физическите проблеми свързани най-често с прекъсване на кабелни трасета или механични повреди по оборудването и поддържащата инфраструктура днес далеч по-често се срещат инциденти свързани с умишлено или непреднамерено увреждане на конфигурацията на комутационната инфраструктура. Тъй като съвременните мултимедийни приложения изискват далеч по-голям обем трафик, който следва да бъде доставен в нужното място с минимално времезакъснение и загуби, прилагането на специализирани системи за перманентен мрежов мониторинг днес придобиват все по-голяма популярност сред мрежовите администратори и архитекти. Важна особеност на съвременното мрежово оборудване, модерните Cloud и Data Center решения, виртуализацията и сигурността на мрежите е все по-силното навлизане на Linux базирани операционни системи в мрежовия хардуер. Операционната система Linux има вече над 26 годишна история и в момента съществуват над 80 добре поддържани дистрибуции, като над 95% от всички Linux дистрибуции са практически напълно безплатни [1]. Ниската финална себестойност от внедряването и поддържането на Linux базирани мрежови операционни системи в съвременния хардуер през последните 4-5 години предизвика сериозен интерес към използването на свободната операционна система и я направи

предпочитан избор сред редица производители на популярен и професионален мрежов хардуер. От друга страна прототипите на добре известните ни приложения създадени преди повече от 30 години, като: ping, traceroute, netstat, netcat са основна градивна част от повечето скриптове за автоматизация и детекция на проблеми в големи корпоративни и потребителски мрежи, което ги прави особено полезни инструменти при реализация на сложни мониторинг системи, каквито са нужни например за реализация на силно разпределени мрежи за интернет на нещата, индустриални системи за контрол и т.н. Настоящата статия представя кратък обзор на основните функционалности на тези програмни инструменти реализирани в операционна система Linux.

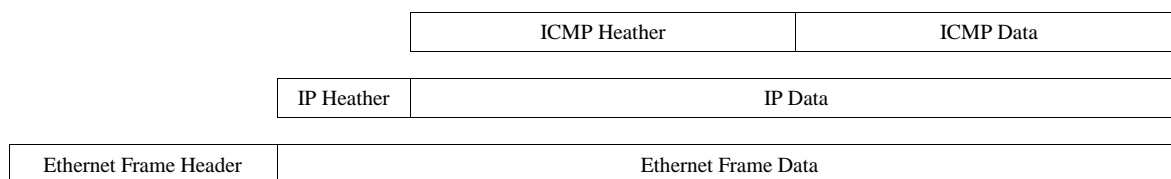
II. НЯКОИ ОСОБЕНОСТИ НА МРЕЖОВИЯ МОНИТОРИНГ ПРИ ИНТЕРНЕТ НА НЕЩАТА

Откриването и отстраняването на проблеми в мрежовата инфраструктура изисква провеждането на регулярни мероприятия свързани с мониторинга и архивирането на мрежовата конфигурация. Успешността на тези мероприятия зависят както от опита на мрежовите администратори така и от наличието и нивото на внедряване на политиките за сигурност и мениджмънт на самата организация управляваща и експлоатираща мрежовите ресурси. Като правило най-елементарната превенция се извършва чрез водене на архивни копия и документация за цялата мрежова архитектура, което позволява бързото възвръщане на нейния активен статус при случай на авария или кибер атака. Болшинството мрежови системи днес поддържат отдалеченото управление и зареждане на конфигурации в мрежовото оборудване, а където това не е възможно поради политики за сигурност или липса на подобна функционалност в самото оборудване, се налага ръчното зареждане и архивиране на управляващи конфигурации на мрежовия хардуер. Като цяло тези проблеми в още по-голяма степен касаят вградените индустриални системи или така нареченият интернет на нещата. Силно разпределената мрежова архитектура в тази насока, както и ниската производителност на крайното оборудване, излизаци от изискванията за свръх ниска консумация и максимално ниска цена често пъти налага имплементацията на ръчно конфигурирани мрежови контролери (например болшинството евтини TCP/IP серийни удължители ползвани с цена под 100USD за свързване на управляващи устройства днес не поддържат IPv6, а редица индустриални приложения работят директно с Dedicated Ethernet канали, като псевдо TCP/UDP/IP стек се реализира на приложно ниво с най-ниска функционалност, отсъствие на групово адресиране, силна фрагментация на пакетите и т.н. [2]). Тези ограничения не позволяват крайното мрежово оборудване да бъде отдалечено конфигурирано, като така за мониторинга на тези вградени системи се налага използването на възможно по-елементарни мрежови приложения, които данни те могат да обработят в реално време. Съществуват естествено и други протоколи подходящи за реализация на разпределени решения за автоматизация и управление със свръх ниска консумация, като например ZigBee, MiniWi и др. Тези специализирани решения обаче не позволяват масовото въвеждане във вече изградена инфраструктура, поради необходимостта от това административния състав да бъде обучен да използва нови типове софтуер и услуги за разгръщане и следене на статуса на мрежата. Именно поради това болшинството производители на подобни вградени в чипа IP контролери се стараят да обезпечат минимална съвместимост със стандартните средства за мониторинг на мрежовия трафик. Така мрежовият администратор на подобна система може да бъде класически IP администратор, който ще използва същите мрежови средства за мониторинг, откриване и отстраняване на възникнали проблеми, каквито се ползват в масовата практика на интернет провайдерите и корпоративните IP мрежи, а при нужда да обедини в управляващи скриптове функциите на стандартни приложения за извършване на по-трудоемки задачи. В повечето случаи воденето на регулярни архивни копия на конфигурацията, използването на Wireshark (Ethereal), преглеждането на журналната

мрежова файлова система, организацията на перманентен мониторинг и анализ на трафика са достатъчни за поддържане на мрежовата инфраструктура в добро активно състояние.

III. ПРОГРАМИ ЗА ПРОВЕРКА НА МРЕЖОВАТА СВЪРЗАНОСТ

Де факто крайното оборудване не разполага със специализиран софтуер за изграждане логическа топология на активните сегменти в мрежата. Обикновено в повечето случаи, такъв софтуер е служебен и се използва от маршрутизаторите, но съществуват и редица програми използвани главно за мониторинг, откриване и отстраняване на проблеми, които могат да бъдат изпълнявани и от клиентски терминали. Болшинството програми и инструменти за контрол на проходимостта между мрежовите сегменти и мрежи са базирани на Internet Control Message Protocol (ICMP). Крайните хостове и шлюзове използват протокола, дефиниран от RFC 792, като протокол за контрол и изпращане на съобщения за диагностика. ICMP съществува в мрежовия слой на модела OSI и в слоя Интернет на модела DoD. Протоколът ICMP се счита за интегрална част от протокола IP, но за разлика от него обезпечава някакъв вид обратна връзка под формата на съобщения за възникнали грешки, което позволява динамично да се управлява скоростта на предаване. Той използва услугите на IP за доставяне на своите съобщения, т.е. ICMP е енкапсулиран в IP и на практика представлява сигнализационен протокол. В този случай в IP хедъра в полето Protocol ще бъде записано числото 1, което индикира, че следващият протокол е ICMP (Фиг. 1.a и 1.b):



Фиг. 1.a. Енкапсулация на ICMP пакета в IP и Ethernet рамката.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type								Code								ICMP Header Checksum															
Identifier																Sequence Number															

Фиг. 1.b. ICMP пакет

- Вида на съобщението се описва в полето "Type".
- Вида на грешките се определя по стойностите в полето "Code".
- Полето "Identifier" задава стойност при необходимост от ехо запитвания и ехо отговори.
- Полето „Sequence Number“ указва поредни стойности за подреждане на ехо запитванията и ехо отговорите.

Хостовете и шлюзовете на местоназначението по принцип трябва да информират хоста-източник за проблеми по доставянето на пакети, да тестват възможността за връзка към този хост, да искат намаляване на скоростта на предаването и др. ICMP има най-различни съобщения, идентифицирани от стойността, съдържаща се в неговото поле за тип и в полето за код, които се използват за информиране на хоста-източник. Тези съобщения дават възможност на хоста-източник да научава за проблеми, които могат да възникнат по мрежата. Едни от най-важните съобщения за грешки касаят: дали търсеното

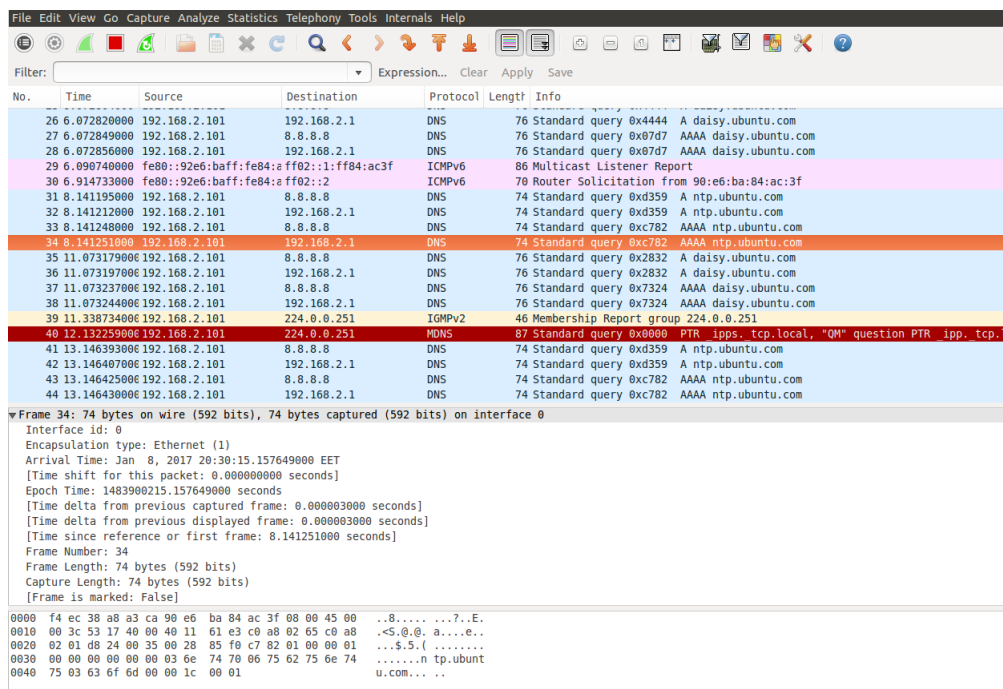
назначение е достъпно, дали има потвърждение за вече установена връзка, изтекло време, проблем с подадения параметър или извършване на препращане на пакета и искане за ехо отговор или повторно искане за отговор, когато TTL полето стане равно на 0, ICMP изпраща съобщение за изчерпано време на препредаване, което е полезно при мрежи с дублирани връзки без активен spanning tree протокол при комуникация между рутери, а при комуникация с потребителски хост грешката означава, че не всички фрагменти на пакета са били получени. Въпреки, че тези съобщения информират хоста за проблеми, ICMP не гарантира тяхното решаване. Подобно на IP, ICMP е протокол без установяване на съединение. Хостовете и шлюзовете могат да изпращат по собствена инициатива ICMP контролни или диагностични съобщения. Ехо-запитването на ICMP е най-често срещаният тип съобщение. Ехо-запитването се използва като диагностично средство, за проверка на възможността за връзка между крайни хостове. Споменаването на небезизвестните програми: ping, traceroute и netstat макар да бъдат умиление у повечето начинаещи мрежови експерти във всъщност стоят в основата на реализацията на автоматизационни скриптове за имплементация на редица политики за мониторинга и откриване на неизправности и проблеми в мрежовата инфраструктура. Названието на вероятно най-популярната програма - **ping** произлиза от звука на сонарите, а не е съкращение. Програмата е разработена през декември 1983г. от Mike Muuss, като свободно използвана програма за проверка на мрежовата свързаност. Приложението ping използва Internet Control Message Protocol (ICMP) като изпраща заявки до отдалечен хост и очаква отговор на тях, програмата извежда статистика за времето необходимо за преминаване на пакетите по правия и обратен път, загубата на пакети, извежда осреднени показатели на тези параметри. ICMP винаги изпраща съобщения за възникнали грешки. Филтрирането на изхода от всички мрежови програми чрез gter ни позволява да извличаме само интересуващата ни информация. Дефакто чрез ping скриптове повечето провайдери обезпечават вътрешен мониторинг на свързаност и проходимост на своите потребители. Пакетът на ICMP е достатъчно малък, за да може да бъде успешно използван в приложения с ограничена памет и бързодействие, и това прави и приложението ping удобно средство за мониторинг на отдалечени мрежови възли и крайни вградени устройства и сензори с частично реализиран IP стек. Друга ценна програма е **traceroute**, тя работи подобно на ping, но дава информация относно пътя по който даден пакет преминава до своето назначение, програмата изисква от всеки рутер по пътя към местоназначението на пакетите да отговори на подадените заявки, като така се проверява времето за преминаване от източника през всички маршрутизатори. За работа с тези програми може да използвате командния ред или програмата Network Tools, която дава единен графичен интерфейс към тях. Реалната полза от тях обаче има при употребата им в скриптове, като например скрипта, който тества за свързаност с определени адреси описани в допълнителен файл. Обикновено адресите са предварително известни, като мрежовият администратор описва детайлно всяко едно устройство във своята мрежа (в примера във файла list.txt).

```
#!/bin/bash
date
cat list.txt | while read output
do
    ping -c 1 "$output" > /dev/null
    if [ $? -eq 0 ]; then
        echo "node $output is up"
    else
        echo "node $output is down"
    fi
done
```

Редица мрежови устройства е възможно да не притежават предварително зададен IP адрес, но да не поддържат неговото автоматично придобиване след включване в мрежовия сегмент чрез протокола Dynamic Host Configuration Protocol (DHCP). Подобни устройства са редица TCP/IP RS232 удължители, като за решение на проблема в подобни случаи устройствата се идентифицират само на ниво Ethernet. За да бъдат достъпни първоначално фабрично известният MAC адрес на контролерите може да бъде въведен в конфигурационен файл, който специално приложение или чрез скрипт да сканира за инсталираните устройства в мрежата изпращайки Ethernet рамки до тях проверявайки за отговор и задавайки им начален IP адрес съобразно техния физически адрес. Това е особен проблем при контролерите с малък обем памет, където реализацията на TCP/IP стека е частична. За произволни експерименти с генериране на пакети и пенетрация на мрежата може да се ползват или специално написани програми позволяващи на ядрото на операционната система да изпраща директно Ethernet рамки или програми, като **packeth** <http://packeth.sourceforge.net/packeth/Home.html>. Друго интересно приложение, което може да се използва за нуждите само на локалната мрежа, поради невъзможността на пакетите да се препредават през маршрутизатори е ARP, използващо протокола Address Resolution Protocol (ARP), във IPv6 се използва функцията Network Discovery (ND) част от протокола ICMP. ARP позволява откриването на физическия адрес на получателя на база неговия IP адрес, често системите за начална инициализация на разнообразни вградени решения използват този протокол, като проверяват само онези върнати пакети касаещи MAC адреси попадащи в определен обхват от адресното пространство на производителя на контролерите. Програмата **netstat** е вероятно една от най-полезните при анализ на работата на мрежовите приложения и интерфейси. Извежданата от нея информация може да се използва за откриване на проблеми на канално ниво или по-горни слоеве при анализа на мрежовите характеристики. Задавайки различни входни параметри на командата тя може да извежда данни за мрежовите съединения (`netstat -a`), таблици с маршрутите (`netstat -r`), статистика на интерфейсите (`netstat -i`), маскираните съединения, мрежовата статистика (`netstat -s`) и принадлежността към групови адреси (`netstat -g`) [3]. Програмата **netcat** е вероятно най-удобният инструмент за тестване и осъществяване на мрежови съединения през TCP/UDP, а също така и за писане на не много натоварени скриптове за работа в мрежа. Програмата е мрежова версия на програмата `cat`, използвана за четене и извеждане на информация от файлове. За използването и в TCP съединение трябва да се стартират две версии на програмата, едната конфигурирана като сървър (`nc -l port_number`) на единия хост, а другата като клиент на отдалечения хост (`nc Server_IP_Address port_number`), който иска да се свърже с нея. За да не бъде затворена сървърната версия на програмата и тя да може да получи ново съединение с друг клиент след приключване на сесията програмата следва да бъде стартирана със следната опция `-k` (`nc -k -l port_number`). Ползването на програмата за UDP обмен става по подобен начин (`nc -l -u 12345` за чакащото съединение приложение и `nc localhost -u 12345` за програмата, която трябва да се свърже с определения сървър). Недостатък на UDP комуникацията с `netcat`, е че при приключване на комуникацията от страна на вече свързан клиент за правилно функциониране на сървърната част тя следва да се рестартира. `Netcat` е мощен инструмент, чрез който може да се обменят файлове и да се пишат доволно сложни скриптове за мрежов мониторинг и автоматизация. Друг полезен инструмент е програмата **tcpdump**, която следва да се пуска в администраторски режим, тъй като позволява извеждане на пълна информация за вървящия по мрежовия интерфейс трафик. Чрез `grep` и пренасочване с `cat` изведените съобщения успешно могат да се използват за анализ на мрежови процеси и разработка на приложения, като данните бъдат записани в лог-файл. Освен тези стандартни програми даващи пълна информация за мрежовия трафик по

интерфейси, сокети и приложения често се налага по-семпло фокусирано наблюдение на мрежовия трафик и извеждане на информация, която директно може да бъде консумирана от мрежовия администратор без допълнителна обработка [4], такива програми са: Nload, iftop, iptraf, nethogs, bmon, tcptrack, bwm-ng, speedometer, PktstatNetwatch и др. Различните програми използват различни механизми, за да агрегират информация за актуалния трафик, някои четат директно файла /proc/net/dev, други използват библиотеката rpsar, за да прихванат директно целия входящ трафик, повечето програми използват ASCII изход на графиките, а някои могат да дават осреднени статистически параметри за отделните съединения, приложения и т.н.

Wireshark (или **Etherreal**) е вероятно най-популярната и ползвана от всички мрежови администратори програма за анализ на мрежовия трафик под Linux (Фиг. 3) и Windows. За правилна работа на софтуера в Linux следва да бъде стартиран с административни привилегии, тъй като осъществява достъп до мрежовите интерфейси на ниско ниво. Съвременната версия на софтуера позволява анализ на трафика, както на IP мрежи (жични и безжични), така и анализ на трафика в други стандартни пакетно ориентирани мрежи, като: ANSI, GSM, H.225, IAX2, ISPU, LTE, MTP3, RTP, RTSP, SCTP, SIP, SMPP Operations, UCP Messages, VoIP Calls, WAP-WSP. За работа със стандартни телекомуникационни мрежови интерфейси, програмата следва да бъде пренасочена да получава данни от OpenSDR софтуер или друг канален интерфейс за прихващане първичните данни по радиоинтерфейса. Данните от криптирани конекции могат да бъдат допълнително подложени на анализ чрез приложения, като airocrack.



Фиг. 3 Общ екранен изглед на програмата Wireshark

IV. ЗАКЛЮЧЕНИЕ

Операционната система Linux изобилства от стандартно вградени в ядрото и редовните инсталации средства за анализ и мониторинг на мрежовите съединения, прослушване на мрежовия трафик и статистика. Редица от тези софтуерни пакети следва да бъдат инсталирани допълнително, но практически всички необходими инструменти са напълно безплатни за използване от всеки мрежов администратор и потребител. Наличните програми могат да бъдат използвани за създаване на многофункционални

скриптове за мониторинг на мрежовия трафик и редовна статистика, а също така и за откриване и отстраняване на неизправности, като проблеми с DHCP, DNS, отпадане на физически съединения и т.н. Редица програми, като `ring` ползващи ICMP удачно се имплементират в частично реализиран IP стек, което прави този метод за мониторинг на отдалечени хостове и устройства особено удобен в контекста на развитието на интернет на нещата. Съществуват и други по-опростени програми за мониторинг на мрежата, като някои от тях извеждат информация и статистика за точно определени интерфейси и съединения, а други като `tcpdump` извеждат абсолютно всички пакети нефилтрирано, като оставят на потребителя възможността да обработи статистически изходните данни със скрипт, особено удобно при реализация на вградени системи с малко оперативна памет работещи в реално време. Съществуват и пенетрационни тестови приложения (предимно написани на C и изискващи прекомпиляция), чрез които администратора може да извършва генериране на готови пакети от файлове или пакети с псевдослучайни стойности на канално и мрежово ниво с произволна информация за адресацията на канално и мрежово ниво. Ето защо доброто познаване на средствата за мрежов мониторинг и тестване на мрежовите съединения следва да залегне по-задълбочено в обхвата на приложно преподаваните дисциплини в областта на IP и мрежообразуването.

ЛИТЕРАТУРНИ ИЗТОЧНИЦИ

[1]. KELLEY, Todd. Linux History. *Ian D. Allen Personal Home Page* [online]. [viewed 11 November 2016].

Available from: <http://www.idallen.com/>

[2]. ПЕТРОВ, Георги, Филип АНДОНОВ и Тодор ДАЧЕВ. *Разработка на приложения с отворени хардуерни платформи*. София: Авангард Прима, 2015. ISBN 978-619-160-506-4. ; Petrov, G., F. Andonov, T. Dachev. *Razrabotka na prilozhenia s otvoreni harduerni platformi*. Sofia: Avangard Prima, 2015, ISBN 978-619-160-506-4.

[3]. MANN, Scott and Mitchell KRELL. *Linux TCP/IP network administration: [one-stop source for Linux network administrators; implementing, managing, and troubleshooting Linux networks; covers Linux routing, file management, directory services, e-mail, security]*. Upper Saddle River, NJ: Prentice Hall PTR, 2002. ISBN 0-13-032220-2.

[4]. MOON, Silver. 18 commands to monitor network bandwidth on Linux server. *BinaryTides Coding, Software, Tech and Reviews* [online]. Apr 4, 2014 [viewed 11 November 2016]. Available from: <https://www.binarytides.com>

Информация за авторите:

гл. ас. д-р Георги Петров, Департамент "Телекомуникации" при магистърски факултет на НБУ, ул. Монтевидео № 21, 2-609, Тел.: 02 8110609, e-mail: gpetrov@nbu.bg

доц. д-р Иван Богомилов, Департамент "Телекомуникации" при магистърски факултет на НБУ, ул. Монтевидео № 21, 2-609, Тел.: 02 8110609, e-mail: ibigomilov@nbu.bg

Contacts:

Assist. Prof. Georgi Petrov, PhD, New Bulgarian university, Department Telecommunication, Sofia, 21 Montevideo St., tel.: 359 2 8110609; e-mail: gpetrov@nbu.bg

Assoc. Prof. Ivan Bogomilov, PhD, New Bulgarian university, Department Telecommunication, Sofia, 21 Montevideo St., tel.: 359 2 8110609; e-mail: ibigomilov@nbu.bg

Дата на постъпване на ръкописа (Date of receipt of the manuscript): 09.01.2017

Дата на приемане за публикуване (Date of adoption for publication): 11.09.2017