O’ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI

MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
NUKUS FILIALI

Telekommunikatsiya injiniringi kafedrasi

Telekommunikatsiya texnologiyalari yo’nalishi

Himoyaga ruhsat etildi

Kafedra mudiri

Seitnazarov K. K.

_____

2017 y. «___»_____

Dasturiy konfiguratsiyalanadigan tarmoqlarning operatsion tizimlari tahlili

mavzusida

BITIRUV MALAKAVIY ISHI

Bitiruvchi : _____ Pirnazarov A. Q.

Rahbar: _____ Bisenbaev J..

Nukus 2017

Mundarija

KIRISH

Bugungi kunda ilm-fan rivojlanishi bilan birga jamiyatimizda axborot kommunikatsiya texnologiyalarining o’rni ham o’sib bormoqda. Buni davlatimizda olib borilayotgan elektron hukumatga o’tish jarayonlarida olib borilayotgan ishlar, kundalik hayotimizga yangidan kirib kelayotgan xizmatlar, masalan elektron to’lov xizmati, yo’l transport qoidalarini bajarilishini kuzatuv kameralari yordamida nazorat qilish tizimining va boshqa bir qator xizmatlarning joriy qilinishi bilan tasdiqlash mumkin. Xizmatlar sonining ko’payishi o’z navbatida foydalanuvchilar sonining ortishiga olib keladi. 2016 yilning III choragidagi ma’lumotlarga ko’ra O’zbekistondagi Internet foydalanuvchilari soni 13 milliondan oshgan. Mamlakatimiz hududining 92% i mobil aloqa bilan qamrab olingan. Magistral telekommunikatsiya tarmog’i xalqaro va tuman darajasida 10 barobar, viloyatlar darajasida 4 barobarga kengaytirilgan [11]. Keng polosali internet xizmatlarini ko’rsatuvchi simli portlardan foydalanish 107,4% o’sgan. Bu esa foydalanuvchilarga keng polosali IPTV, videotelefoniya, yuqori tezlikdagi internet, HDTV-kanallar va boshqa xizmatlarni ko’rsatish imkonini yanada kengaytirmoqda [12]. 2017 yil 1-yanvar holatiga xalqaro internet tarmog’iga kirish 59 GbitG’s tezlikga etkazilgan. Dasturiy ta’minot ishlab chiqarish bilan shug’ullanuvchi sub’ektlar soni 360 tani, davlat axborot resurslari soni 499 tani tashkil etgan [13].

Kompyuter tarmoqlarining zamonaviy holati va rivojlanish tendentsiyalari, shuni ko’rsatdiki, an’anaviy texnologiyalarga asoslangan tarmoqlarning o’tkazish qobiliyati, tezkorligining o’sish potentsiali amalda o’z chegarasiga etgan. Bu marshrutlash vaqti harajatlari, tarmoqni konfiguratsiyalash qiyinchiliklari va unda oqimlarni boshqarish bilan bog’liq, asosan yuqori tezlikli global tarmoqlar va ma’lumotlarga ishlov berish markazlari uchun xizmat sifati politikalarida yangi talablarni hisobga olgan holda, tarmoqlarni virtuallashtirishga talabning oshishi bilan bog’liq.

Virtuallashtish deyilganda umumiy tarmoq resurslarida mustaqil xizmat ko’rsatish sifati politikalariga ega birnechta mantiqiy izoltsiyalangan tarmoqlarni ifodalash tushuniladi.

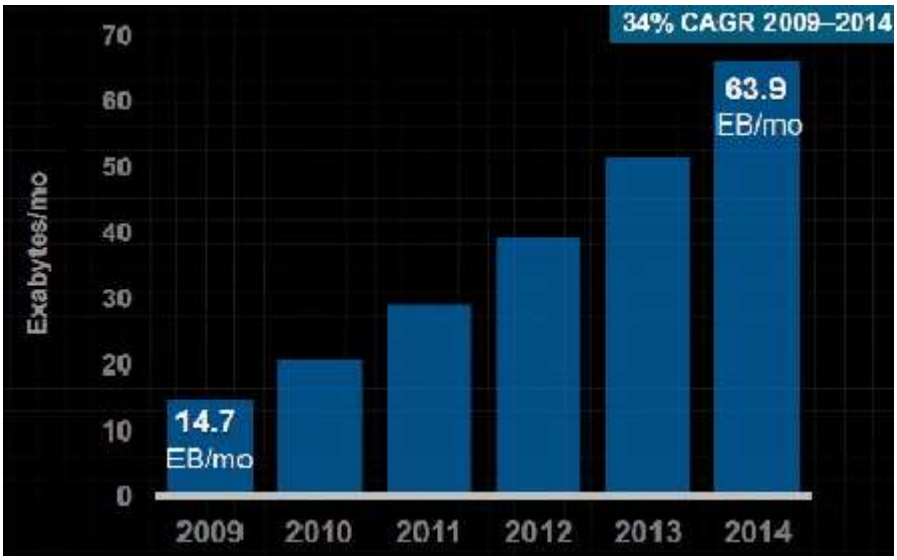
Yuqorida ko’rsatilgan muammolarga javob sifatida 2006 yili dasturiy-konfiguratsiyalanadigan tarmoqlar kontseptsiyasi yuzaga keldi (shundan buyon intensiv rivojlanmoqda). Bu kontseptsiya tarmoqlarda marshrutlashni tezlashtirish, konfiguratsiyalash, virtuallashtirish, xizmat ko’rsatish sifatini sozlash qulayligini oshirish imkoniyati beradi, lekin qo’shimcha tadqiqotlar va ishlanmalarni talab qiladi, xususan, tarmoq kommutatorlari apparaturasini tashkil etish, tarmoqni boshqarish uchun dasturiy ilovalar va ularni bajarish uchun platformalar sohasida o’z echimini kutayotgan muammolar mavjud.

1. DASTURIY KONFIGURATSIYALANADIGAN TARMOQ YARATILISHI VA UNING ARXITEKTURASI

1.1. An’anaviy kompyuter tarmoqlaridagi muammolar

60-70 yillarda birinchi ARPANET tarmog’i yaratilishi bilan an’anaviy kompyuter tarmoqlari va Internet arxitekturasi shakllangan. Lekin shundan beri kompyuter tarmoqlarini (KT) foydalanish sohasida son va sifat jihatdan jiddiy o’zgarishlar ro’y berdi. Bugungi kunda KT arxitekturasi eskirgan va axborot texnologiyalari rivojlanishining yangi tendentsiyalariga va jamiyatning zamonaviy talablariga doim ham tez va samarali javob bera olmaydi.

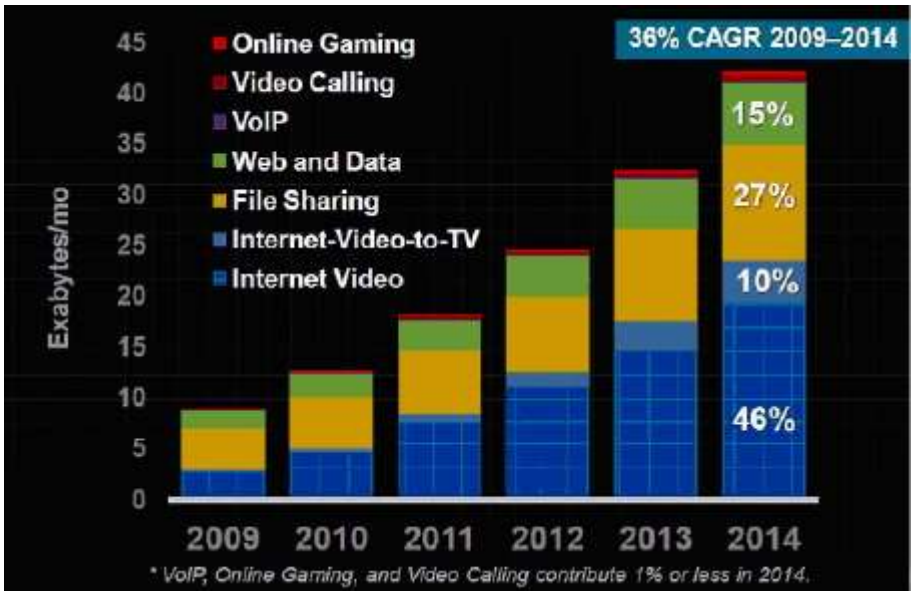
Cisco kompaniyasining 2009-2014 yillardagi tadqiqotlari bo’yicha 2014 yilda global internet-trafigi to’rt marotaba oshadi va 767 eksabaytga (1 eksabayt q 10^{18} bayt) etadi. Tadqiqot diagrammasi 1-rasmda berilgan.



1-rasm. Internet global trafigining o’sishi (oyiga eksabayt)

Bunda trafik strukturasi va dinamikasi o’zgaradi. Prognoz qilinayotgan trafik strukturasi 2-rasmda ko’rsatilgan. O’sishning asosiy omili video-trafik bo’lmoqda. 2014 yilda global foydalanuvchi internet-trafikdagi uning ulushi 91% ni tashkil etadi. Tarmoq o’tkazish qobiliyatining va Internetda ma’lumot uzatish tezligining kengayishi, shuningdek, yuqori aniqlikdagi televidenie (HDTV) va hajmli televidenie (3DTV) ommalashishi 2009 dan 2014 yilgacha IP-trafikning to’rt

marotaba oshishida asosiy omil hisoblanadi. 2014 yilda 3D va HD videotrafiging hajmi Internetdagi umumiy foydalanuvchi trafiging 42% ini tashkil etadi. Veb-konferentsiyalar segmenti hammasidan tez rivojlanmoqda: 2009 yildan 2014 yilgacha davrda 180 marotaba o'sgan.



2-rasm. Prognoz qilinayotgan trafik strukturasi

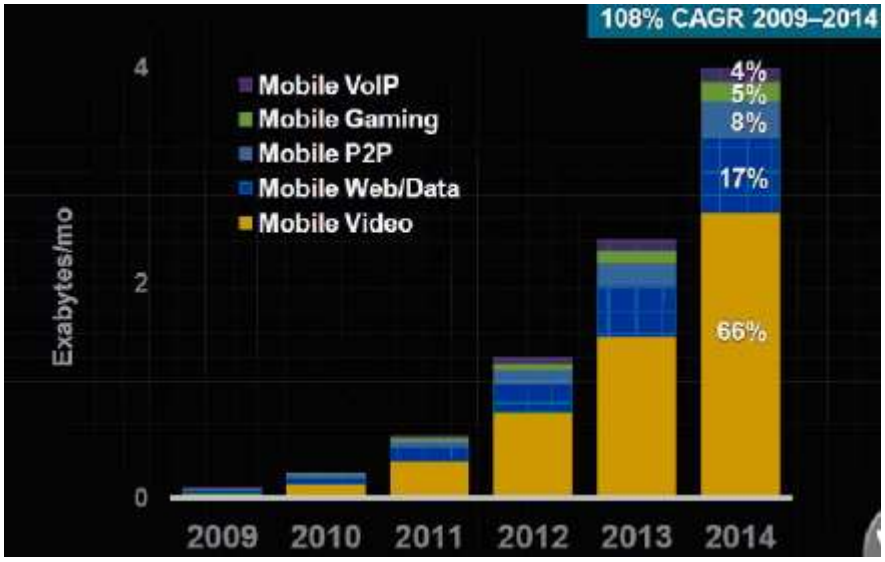
Bu raqamlar shuni ko'rsatadiki, tarmoqda trafikni boshqarishning mavjud usullari va vositalaridagi zamonaviy aloqa kanalining o'tkazish imkoniyati nihoyasiga etmoqda. O'tkazish qobiliyati o'sishining mavjud shiddati foydalanuvchilarning o'suvchi talablarini qanoatlantirish holatida bo'lmaydi.

Mobil qurilmalarning turi va soni o'sishi, har hil simsiz aloqa texnologiyalari (WiFi, 3G, WIMAX, LTE) rivojlanishi, mobil servislar soni o'sishi, bugungi kunda simsiz texnologiyalar asosidagi kompyuter tarmoqlari foydalanuvchilari soni turg'un aloqa foydalanuvchilar sonidan oshishiga olib keldi, bitta foydalanuvchiga to'g'ri keladigan mobil terminallar soni esa rivojlangan davlatlarda uchtani tashkil etadi.

Lekin mobil simsiz tarmoqlar bugungi kunda ikkita qarama-qarshi tendentsiyalar bilan to'qnash kelmoqda. Mobil terminallarning hisoblash quvati oshishi, ularda ishlovchi ilovlarning hisoblash sig'imi oshishiga olib kelmoqda. Bu o'z navbatida mobil aloqa kanallarining o'tkazish qobiliyatiga talabning ortishga

olib keladi. Bugun mobil trafik hajmi geometrik progressiya bo'yicha o'smoqda, trafik turi esa yana ham turli hil bo'lib bormoqda.

2009-2014 yillar mobaynidagi Cisco tadqiqotlari ma'lumotlari bo'yicha, ma'lumot uzatish bilan bog'liq global mobil trafik hajmi 2014 yilda 39 marotaba o'sgan va oyiga 3,5 eksabayt yoki yiliga 42 eksabaytni tashkil etadi. Mobil trafik o'sish tendentsiyasi 3-rasmda ko'rsatilgan.



3-rasm. Mobil trafik o'sishi va uning turlari bo'yicha taqsimlanishi

Bir vaqtning o'zida kompyuter tarmoqlariga yuklamaning miqdoriy ko'rsatkichlari o'sishi bilan birga tarmoqni boshqarish qiyinchiligi ortdi. Echiladigan masalalar ro'yxati sezilarli kengaydi, ularning ahamiyati va nozikligi tarmoqlarning havfsizlik va ishonchliligiga talablarni oshirdi.

Tarmoqlar kommutatorlar, marshrutizatorlar va boshqa qurilmalarni foydalangan holda quriladi va ular ko'p sonli murakkab taqsimlangan protokollarni amalga oshirganligi sababli haddan tashqari murakkab bo'lib bormoqda. Bugungi kunda IETF tomonidan standartlashtirilgan faol ishlatilayotgan protokollar va ularning versiyalari soni 600 dan ortiqni tashkil etadi.

Bunday sharoitlarda tadqiqotchilar ularga zarur eksperimentlarni ishlab turgan tarmoqlarda o'tkaza olmaydilar, operatorlar o'zlarining foydalanuvchilari uchun yangi xizmatlarni tez kirita olmaydilar, tarmoq qurilmalari ishlab chiqaruvchilari

esa buyurtmachilarning talablarini qondirish uchun innovatsiyalarni shunchalik tez joriy eta olmaydilar.

Shunday qilib, an’anaviy tarmoqlarda echimi mavjud arxitekturaga o’zgartirish kiritishni talab qiladigan bir qator muammolar to’plangan:

1. Ilmiy-texnik muammolar – bugun kompyuter tarmoqlari singari murakkab ob’ektlarni nazorat qilish va holatini ishonchli oldindan ko’rish mumkin emas.
2. Ijtimoiy – kundalik hayotimizda biz barchamiz ko’proq va yana ko’proq Internetga tayanamiz. Lekin, ma’lumotlar shu bilan birga shaxsiy ma’lumotlar havfsizligi kafolatlanmagan, Internet tashqi hujumlarga bardosh bera olmaydi.
3. Iqtisodiy – tarmoqlar qimmat va xizmat ko’rsatish murakkab, yuqori kvalifikatsiyali mutaxassislarni talab qiladi.
4. Rivojlanishda muammolar – zamonaviy tarmoqlar arxitekturasida innovatsiyalar kiritishga, eksperimentlashga, yangi servislarni yaratishga barer mavjud.
5. Kompyuter tarmoqlari arxitekturasi uni qurish tamoyillarini qayta ko’rib chiqishni talab qiladi.

1.2. Dasturiy konfiguratsiyalanadigan tarmoqlar arxitekturasi

Dasturiy konfiguratsiyalanadigan tarmoq (DKT) – bu kompyuter tarmoq arxitekturasini qurishga yangicha yondashuvdir, bunda boshqaruv funktsiyalari kontroller deb nomlanadigan alohida qurilmaga o’tkazilishi hisobiga tarmoqni boshqarish pog’onasi (BP) va ma’lumotlarni uzatish pog’onasi (MUP) ajratiladi. Boshqaruv funktsiyalari an’anaviy tarmoqlarda marshrutizatorlar va kommutatorlarda amalga oshirilar edi.

Bunday ajratish hisobiga tarmoqning holatini nazorat qilish kontrollerda mantiqan markazlashgan. Bundan tashqari bunday yondashuv boshqarish pog’onasini ma’lumotlar uzatish pog’onasining fizik tarmoq infrastrukturasidan

mavhumlashtirish imkoniyatini beradi. BP va MUP o’rtasida munosabat yagona unifikatsiyalangan ochiq interfeys yordamida amalga oshiriladi.

DKT yondashuvining asosiy g’oyasi Stenford (Nik Makkeon) va Berkli (Skott Shenker) universitetlari mutaxassislari tomonidan 2006 yilda ta’riflangan va akademik muhitda, etakchi tarmoq qurilmalari ishlab chiqaruvchilari hamjamiyatida va yirik IT-kompaniyalari tomonidan keng qo’llab-quvvatlanishiga erishilgan [3].

DKT yondashuviga IT-kompaniyalari tomonidan qiziqish bildirilishiga quyidagilar sabab bo’lgan:

- ✓ Tarmoq qurilmalari samaradorligini 25-30% ga oshirish imkoniyati;
- ✓ Tarmoq ekspluatatsiyasi uchun harajatlarni 30% gacha kamaytirish;
- ✓ Dastur yozish hisobiga tarmoqni boshqarish moslashuvchanligi oshishi;
- ✓ Xavfsizlikning sezilarli oshishi;
- ✓ Yangi servislarni dasturiy yaratish va ularni tarmoq qurilmalariga tezkor yuklash imkoniyati.

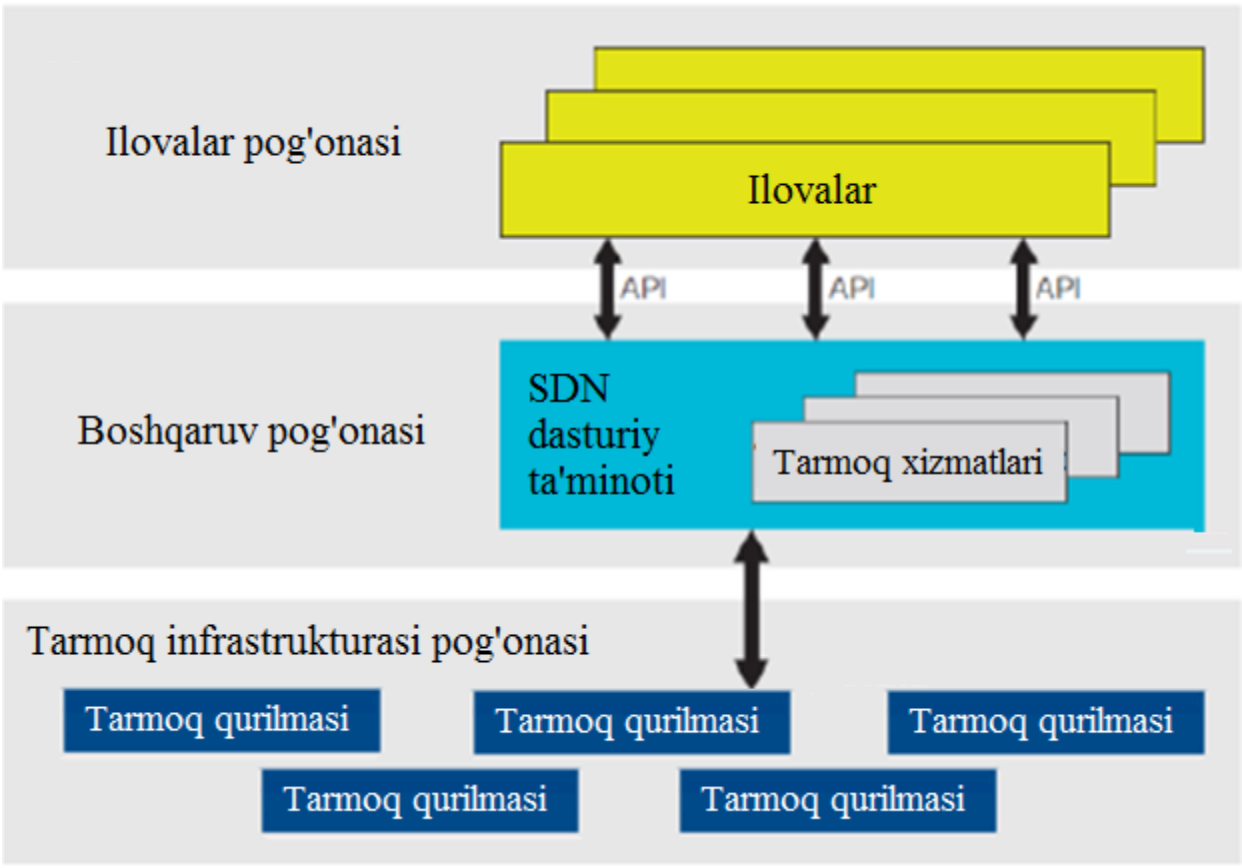
Bu yondashuvning joriy etilishi eng avvalo, ma’lumotlarga ishlov berish markazlari (data-markazlar) tarmog’i, korporativ tarmoq, WAN va uy tarmoqlariga katta ta’sir ko’rsatishi kerak.

DKTni qurishda asosiy g’oyalar quyidagilardir:

1) Ma’lumotlarni uzatish sathi va boshqarish sathini ajratish.
2) Ma’lumotlarni uzatish sathi va boshqarish sathlari o’rtasida yagona, unifitsirlangan, etkazib beruvchidan mustaqil interfeys.
3) Ma’lumotlarni boshqarish sathi mantiqan markazlashgan.
4) Tarmoq fizik resurslarini virtuallashtirish.
5) DKT arxitekturasi uchta sathdan iborat [3] (4-rasm).
6) Tarmoq infrastrukturasi sathi o’z ichiga tarmoq qurilmalarini (kommutatorlar, marshrutizatorlar) va ma’lumotlar uzatish kanallarini qamrab olgan.
7) Boshqarish sathi, tarmoqni global taqdim etishni qo’llab-quvvatlaydigan va kuzatadigan sath. Global taqdim etish deyilganida tarmoq topologiyasi

va tarmoq qurilmalarining holati tushuniladi. Boshqarish sathi tarmoq ilovalari uchun dasturiy interfeys (API) taqdim etadi.

8) Tarmoq ilovalari sathi, tarmoqda ma’lumotlar oqimini boshqarish, xavfsizlikni boshqarish, trafikni monitoring qilish, servis sifatini boshqarish va siyosatni boshqarish va boshqa har xil tarmoqni boshqarish funktsiyalarini amalga oshiradi.



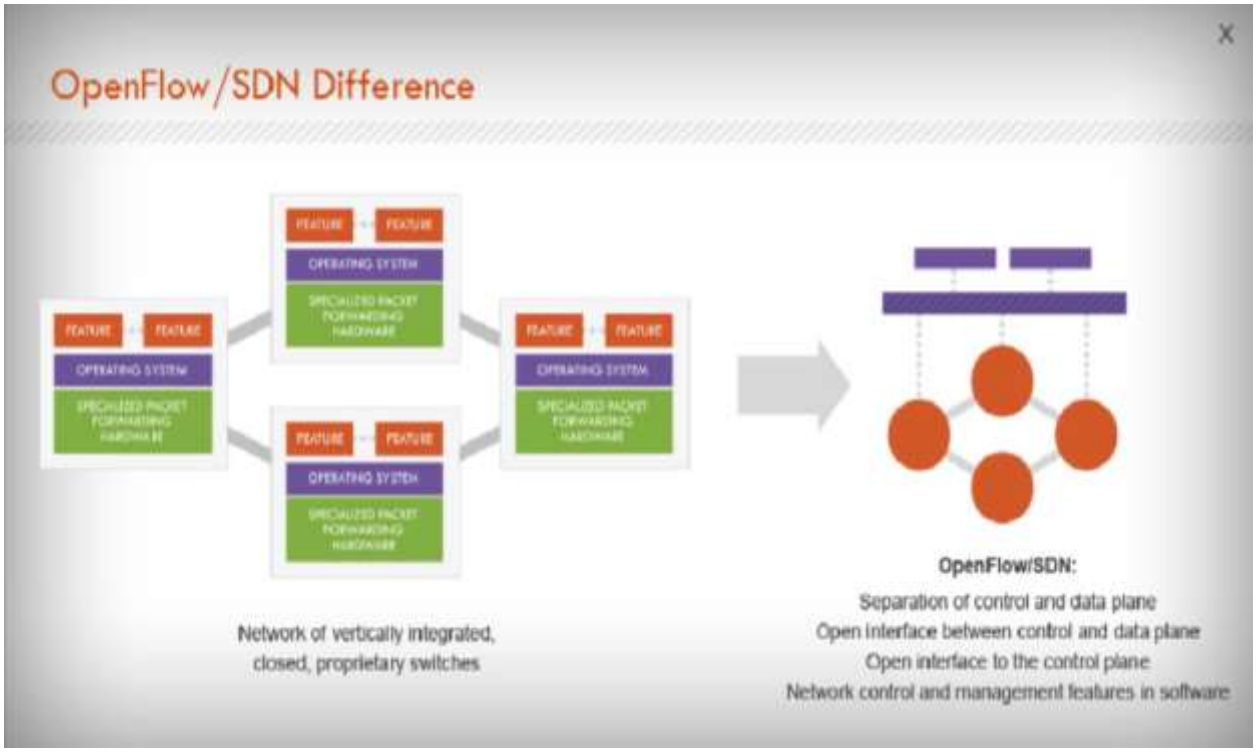
4-rasm. Dasturiy-konfiguratsiyalanadigan tarmoqlar arxitekturasi

Zamonaviy marshrutizatorlar asosiy ikki masalani echadi: ma’lumotlarni uzatish (forwarding) – kirish portidan ma’lum chiqish portiga paketni siljitish va ma’lumotlarni boshqarish – paketga ishlov berish va marshrutizatorning joriy holatiga asoslanib, uni qaerga marshrutlash haqida echim qabul qilishdan iborat. Shunday qilib, barcha tarmoq doirasida ma’lumotlar uzatish vositalaridan (aloqa liniyasi, kanal hosil qiluvchi qurilmalar, marshrutizatorlar va kommutatorlar)

tashkil topgan ma’lumotlar uzatish pog’onasi va ma’lumotlar uzatish vositalari holatini boshqaruv pog’onasini ajratish mumkin.

Marshrutizatorlarning rivojlanishi bu ikki sathlarni bir-biriga yaqinlashuvi va “bir bo’lib ketishi”, apparatli tezlatish, dasturiy ta’minotni takomillashtirish va har bir paketni marshrutlash bo’yicha echim qabul qilish tezligini oshirish uchun yangi funktsional imkoniyatlarni joriy etish yo’lidan bordi. Lekin bunda boshqaruv sathi murakkab taqsimlangan marshrutlash algoritmlari va tarmoqni konfiguratsiyalash va sozlash bo’yicha tushunarsiz yo’riqnomalar sababli etarlicha primitiv bo’lib qoldi. Qayd etish kerak, marshrutizatorlarning boshqaruv pog’onasini amalga oshiradigan dasturiy ta’minot ishlab chiqaruvchilar, tadqiqotchilar va tarmoq operatorlari uchun proprietar va yopiq qoldi.

DKT yondashuvida boshqaruv pog’onasi va ma’lumotlar uzatish pog’onasini ajratish taklif qilindi. DKTda pog’onalarni ajratish sxemasi (5-rasm) da keltirilgan.



5-rasm. DKT arxitekturasi va an’anaviy KT arxitekturasini solishtirish

Shunday qilib, DKT arxitekturasi va taqdim etilayotgan markazlashgan yondashuv ma’lumotlar uzatish boshqaruvi taqsimlangan an’anaviy tarmoqlarga nisbatan quyidagi afzalliklarni taklif etadi:

- ✓ Tarmoqni boshqarishning *moslashuvchanligi va dasturlanganligi*, yangi ilovalar yaratish yoki mavjudlarini modifikatsiyalash hisobiga tarmoqni boshqarishni modifikatsiyalash imkoniyatini sezilarli soddalashtirish, boshqarishni avtomatlashtirish va tarmoqlarni administratsiyalash.
- ✓ Tarmoqni boshqarishning adaptivligi, ya'ni ishlash shartlari o'zgarishini inobatga olib, real vaqt rejimida tarmoq holatini o'zgartirish va unga moslashish imkoniyati, yangi tarmoq ilovalari va servislari yaratish hisobiga tarmoq foydalanuvchilari talablari o'zgarishiga moslashish imkoniyati. Tarmoq ilovalarini ishlab chiqishga barcha tarmoqni qo'lda qayta konfiguratsiyalashga nisbatan sezilarli kam vaqt sarflanadi.
- ✓ Qurilmaga va tarmoq qurilmalari ishlab chiqaruvchilarining yopiq dasturiy ta'minotiga bog'lanib qolmaslik.
- ✓ BP va MUP ni mustaqil kengaytirish imkoniyati
- ✓ Tarmoq qurilmalari strukturasi va mantig'i soddalashadi, shu sababli endi ularga katta miqdordagi protokollar va standartlarga ishlov berish talab etilmaydi, faqatgina kontrollerdan olingan yo'riqnomani bajarishadi.
- ✓ "Routerlar miyasi"ning kontrollerga butunlay ko'chirilganligi sababli kommutatorlar va tarmoq infrastrukturasining narxi sezilarli tushadi.

Lekin DKT arxitekturasida quyidagicha kamchiliklar kuzatiladi:

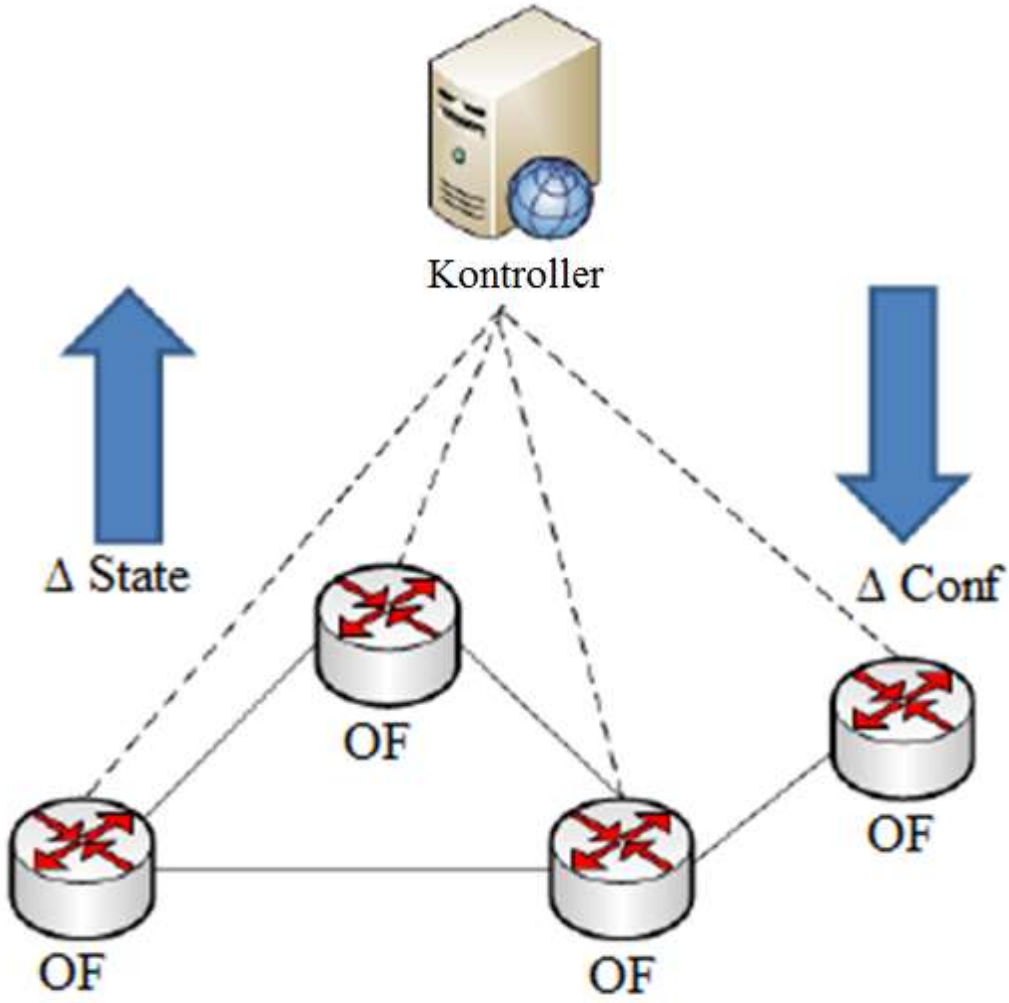
Ishonchlilik muammosi: Kontroller tarmoq ishlashni rad qilishining potentsial nuqtasidir.

1. Markazlashgan kontrollerga mo'ljallangan boshqaruvchi oqimning miqdori, tarmoqdagi kommutatorlar sonini proportsional oshiradi;
2. Yangi oqimlarning ulanish vaqti tarmoq o'lchami oshishi bilan sezilarli o'sishi mumkin.

Unumdorlik muammosi: Tarmoqning unumdorligi to'g'ridan-to'g'ri kontroller unumdorligi va uning fizik imkoniyatlariga bog'liq.

1.3. OpenFlow texnologiyasi

Dasturiy konfiguratsiyalanadigan tarmoqlar yondashuvining eng istiqbolli va rivojlanayotgan realizatsiyalaridan biri OpenFlow texnologiyasi hisoblanadi. Uning asosiy hujjati OpenFlow spetsifikatsiyasi hisoblanadi, unda OpenFlow-tarmoqlarining asosiy komponentlari, ishlash tamoyillari va komponentlar aloqalari ifodalangan (OpenFlow protokoli). Spetsifikatsiya uchun standartlashtiruvchi tashkilot ONF – Open Networking Foundation hisoblanadi. 6-rasmda OpenFlow tarmog'ining asosiy komponentalari ko'rsatilgan [4].



6-rasm. OpenFlow tarmog'ining asosiy komponentlari

Spetsifikatsiyaga mos, OpenFlow tarmoqlari asosiy komponentlari quyidagilar:

- ✓ kontroller, o'z ichiga oladi:

1) tarmoq operatsion tizimi

2) tarmoq ilovalari

- ✓ OpenFlow kommutator;
- ✓ kontroller va kommutator o’rtasidagi himoyalangan kanal;
- ✓ OpenFlow protokoli.

OpenFlow tarmog’i ishlashining umumiy tamoyili quyidagicha: har bir OpenFlow kommutator kontroller bilan himoyalangan kanal o’rnatadi, uning yordamida kontroller ularni boshqaradi. Kommutator va kontroller o’rtasidagi aloqa OpenFlow protokoli xabarlari yordamida amalga oshiriladi. Kontroller tarmoqda elementlar holatining o’zgarishi to’g’risida axborot oladi, uning asosida a tarmoq qurilmasini konfiguratsiyalaydi, tarmoq infrastrukturasini va tarmoqda ma’lumotlar oqimini boshqaradi.

2. DASTURIY KONFIGURATSIYALANADIGAN TARMOQNING ASOSIY KOMPENENTALARI

2.1. OpenFlow kommutator

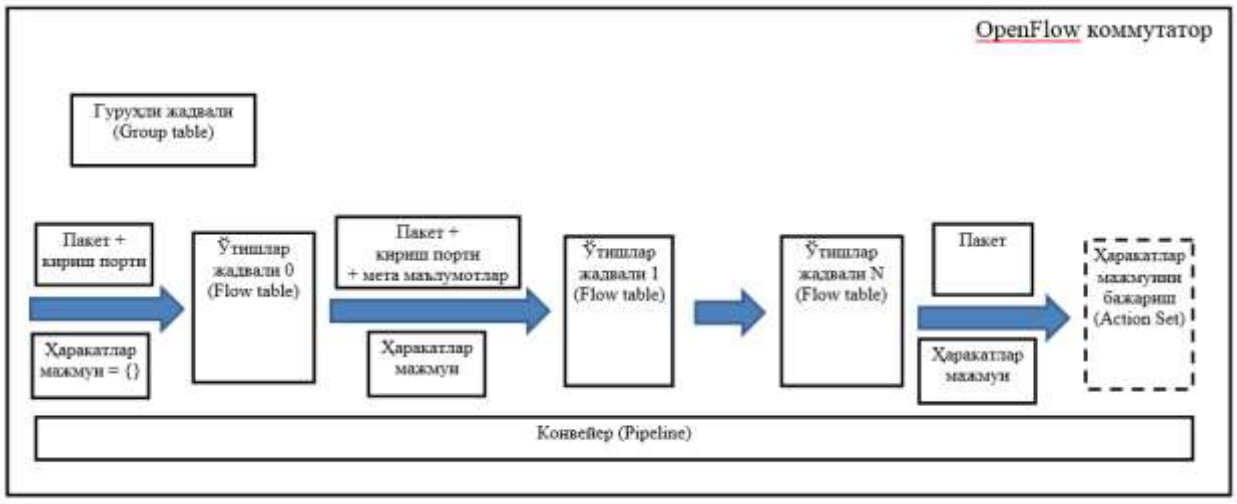
Kommutator tuzilishi. OpenFlow 1.3 versiyasi spetsifikatsiyasiga ko'ra kommutator bir yoki undan ko'p oqimlar jadvali (flow table), guruhli jadval (group table) dan iborat va masofadagi kontroller bilan aloqa uchun himoyalangan kanalni (Secure channel) qo'llab-quvvatlaydi.

Kommutatordagi har bir oqimlar jadvali yozuvlar jamlanmasidan (flow entries) iborat, uni biz oqimlar haqida yozuvlar yoki qoidalar deb ataymiz. Oqim haqida har bir yozuv (qoida) belgilar maydoni (match fields), hisoblagichlar (counters) va yo'riqnomalardan (instructions) tashkil topgan (7-rasm).

Tekshirish maydoni	Hisoblagichlar	Yo'riqnomalar

7-rasm. OpenFlow kommutatorning oqimlar jadvalida qoida tuzilishi

OpenFlow kommutatorining ishlash mexanizmi oddiy bo'lib, har bir kelib tushgan paket kommutator konveyeriga kelib tushadi (8-rasm). Har bir paketda sarlavha (ma'lum uzunlikdagi bitli qator) "kesib olinadi". Bu bitli qator uchun oqimlar jadvalidan birinchisidan boshlab belgilar maydoni ko'proq paketning sarlavhasiga mos keladigani qidiriladi. Agar topilsa, topilgan qoidada ko'rsatilgan yo'riqnomalar majmui bo'yicha paket va uning sarlavhasi ustidan o'zgartirishlar amalga oshiriladi. Har bir jadval yozuvi bilan qo'shilgan yo'riqnomalar paketni uzatish bilan bog'liq, paketning sarlavhasini o'zgartirish bilan bog'liq, guruhli jadvalda ishlov berish bilan bog'liq, paketni kommutatorning ma'lum portiga uzatish bilan bog'liq harakatlarni ifodalaydi. Ishlov berish konveyeri yo'riqnomalari keyingi ishlov berish uchun keyingi jadvalga paketni uzatish va jadvallar o'rtasida axborot (metama'lumot ko'rinishida) uzatish imkoniyatini beradi. Yo'riqnomalar shuningdek, turli statistikalarni yig'ish uchun ishlatilishi mumkin bo'lgan hisoblagichlarni modifikatsiya qilish qoidasini aniqlaydi.



8-rasm. Ishlov berish konveyeridan o’tayotgan paket oqimi

Agarda birinchi jadvalda kerakli qoida topilmasa paket inkapsulyatsiyalanadi va kontrollerga uzatiladi, u mazkur turdagi paketlar uchun mos keladigan qoida shakllantiradi va uni kommutatorga (yoki u tomonidan boshqariladigan kommutatorlarga) o’rnatadi yoki paket (kommutator konfiguratsiyasiga bog’liq) yuqotilishi mumkin.

Yuqorida qayd etilganidek, oqim to’g’risida yozuv paketni ma’lum portga yo’naltirishga buyruq bo’lishi mumkin. Bu odatda fizik port, lekin bu kommutator bilan aniqlangan virtual port yoki protokol spetsifikatsiyasi bilan aniqlangan rezerv qilingan port bo’lishi ham mumkin. Rezerv qilingan virtual portlar yo’naltirishning umumiy harakati aniqlab beradi, misol uchun, kontrollerga keng tarqatiladigan rasso’lkalarni jo’natish yoki OpenFLow-usullar bilan qo’llanilmaydigan yo’naltirishlar, ya’ni kommutatorning “odatdagi” ishlov berishi. Kommutator bilan aniqlangan virtual portlar kanallarning birlashtirish guruhlarini, tunnelarni yoki qayta aloqa interfeyslarini aniqlab berishi mumkin.

Guruhlar jadvali guruhlar haqida yozuvlardan iborat: har bir guruhli yozuv guruh turiga bog’liq maxsus semantikali harakatlar konteynerlari ro’yxatidan iborat. Bitta yoki birnechta konteynerlardagi harakatlar guruhga uzatilayotgan paketlarga qo’llaniladi.

Kommutatorning oqimlar jadvalida qoidalarni o’rnatish, yangilash va o’chirish kontroller tomonidan amalga oshiriladi. Qoida reaktiv (kelgan paketga javob sifatida) yoki proaktiv (oldindan, paketlar kelgunga qadar) o’rnatilishi mumkin.

Ma’lumotlarni boshqarish alohida paketlar darajasida emas, balki paketlar oqimi darajasida amalga oshiriladi. OpenFlow kommutatorda qoida kontroller ishtirokida faqat birinchi paket uchun o’rnatiladi, keyin oqimning qolgan paketlari uni foydalanadi.

Kommutator turlari. OpenFlow protokolini qo’llab-quvvatlaydigan kommutatorlarni ishlashi bo’yicha ikki turga ajratish mumkin: OpenFlow-only va OpenFlow-hybrid [4].

OpenFlow-only – bu faqat OpenFlow operatsiyalarni qo’llab-quvvatlaydigan kommutatorlar hisoblanadi. Bunday kommutatorlarda barcha paketlar OpenFlow konveyerda ishlov beriladi va boshqa usul bilan ishlov berilishi mumkin emas.

OpenFlow-hybrid – bu kommutatorlar bir vaqtning o’zida OpenFlow operatsiyalarni ham, odatdagi Ethernet-kommutatsiya operatsiyalarini ham, ya’ni an’anaviy L2 Ethernet-kommutatsiyani, VLAN, L3 marshrutlashni, ACL va QoS ishlov berishni qo’llab-quvvatlaydi. Bu kommutatorlar yo OpenFlow konveyeriga yo odatdagi ishlov berish konveyeriga marshrutlab trafikni klassifikatsiyalash mexanizmini ta’minlashi kerak. Masalan, paketni qanday yo’l bilan ishlov berish yoki barcha paketlarni OpenFlow konveyerga jo’natishga echim qabul qilish uchun kommutator VLAN teg yoki paketning kirish portini foydalanishi mumkin. Bu klassifikatsiya mexanizmi OpenFlow spetsifikatsiyasi doirasidan tashqari hisoblanadi. OpenFlow-hybrid kommutatorlari shuningdek NORMAL va FLOOD virtual portlari orqali OpenFlow konveyeridan oddiy konveyerga paketlarni uzatishga ruxsat etadi.

Amalga oshirilishi bo’yicha kommutatorlar dasturiy va apparat bo’lib ajratiladi. 1-jadvalda OpenFlow protokoli bo’yicha ishlashni qo’llab-quvvatlaydigan kommutatorlarning mavjud dasturiy amalga oshirilganlari keltirilgan. Bugangi kunda etakchi tarmoq qurilmalari ishlab chiqaruvchilaridan

(NEC, HP va boshqa) OpenFlow protokolini qo'llovchi ko'pgina gibrid kommutatorlar mavjud.

1-jadval. Mavjud OpenFlow kommutatorlarning dasturiy amalga oshirilishi

Nomi	of11 softswich	LINC switch	of13 softswich	Openv Switch	ofss
Ishlab chiquvchi	CPqD	Infoblox	CPqD	Nicira	TrafficLab, Ericsson Research, Hungary
Litsenziya		Apache 2 license	BSD license	Apache 2 license	
Havola	https://github.comG'CPqDG'of11soft switch.git	https://github.comG'Flow ForwardingG' LINC-Switch	https://github.comG'CP qDG'ofsoftswitch13	http://openvswitch.orgG'	https://github.comG'Traffi cLabG'ofss
Protokol	OF 1.1	OF 1.2, OF 1.3, OF-config 1.1.1	OF 1.3	OF 1.2	OF 1.0
Dasturlash tili	CG'C++	Erlang	C++		C++
OT	Linux	Linux	Linux	Linux	Linux

Apparat kommutatorlar OpenFlow protokolining faqat 1.0 versiyasini va bitta oqimlar jadvalini qo'llab-quvvatlaydi. Dasturiy kommutatorlar barcha versiyalarni va ko'p jadvallarni qo'llab-quvvatlaydi.

OpenFlow kommutatorning quyidagi asosiy funktsiyalarini ajratish mumkin:

1. Kontroller bilan himoyalangan kanal yoqish, o’rnatish va qo’llab-quvvatlash;
2. Kontrollerga portlar holati o’zgarishi haqida axborot berish;
3. Kontrollerga oqim yozuvi o’chirilganligi haqida axborot berish;
4. Status o’zgarishi haqida ogohlantirish;
5. Xatolik haqida ogohlantirish;
6. Yangi ma’lumotlar oqimi uchun qoida o’rnatishga talabnoma berish.

2.2. Himoyalangan aloqa kanali

Himoyalangan kanal kommutator bilan kontroller o’rtasida xabarlar almashish uchun ishlatiladi. Odatiy kontroller ko’plab OpenFlow kanallarni boshqaradi. OpenFlow kommutator bir kontrollerga bir OpenFlow kanalga yoki ishonchlilikni oshirish uchun har xil kontrollerga ko’plab kanallarga ega bo’lishi mumkin (1.3 versiya spetsifikatsiyasida).

OpenFlow kontroller odatda bitta yoki birnechta tarmoqlarda OpenFlow kommutatorlarni masofadan boshqaradi. Kommutator va kontroller o’rtasida OpenFlow kanal TLS yoki oddiy TCP ni foydalanib o’rnatiladi. Parallellikni ishlatish uchun OpenFlow kanal birnechta tarmoq ulanishlaridan tashkil topishi mumkin (1.3 versiyasida – qo’shimcha ulanishlarni ishlatgan holda). OpenFlow kommutator OpenFlow kontrollerga ulanishni quyidagicha amalga oshiradi.

Kommutator kontroller bilan ulanish o’rnatishi uchun kontrollerning IP adresi va porti aniqlanishi kerak. Agar bu ma’lumotlar kommutatorga ma’lum bo’lsa, u kontrollerga standart TLS yoki TCP ulanish amalga oshiradi.

Qachonki, OpenFlow ulanish o’rnatilganida har bir taraf uzatuvchi (kommutator) qo’llab-quvvatlaydigan eng katta raqamli protokol bilan OFPT_HELLO xabarini tezda jo’natishi kerak. Bu xabardan keyin kontroller keng kichik versiyani hisoblaydi. Agar bu versiya kontroller tomonidan qo’llab-quvvatlansa ulanish davom etadi. Aks holda OFPT_ERROR xabari jo’natiladi.

Ulanishning to’xtatilishi. Agarda kommutator echo so’rovnomalarning taymauti natijasida, TLS sessiyalar taymauti va shunga o’xshash natijasida barcha kontrollerlar bilan kontaktni yo’qotsa, kommutator tezda kommutatorning amalga oshirilishi va konfiguratsiyasiga bog’liq yo “failsecuremode” yoki “failstandalonemode” rejimiga o’tishi kerak. “failsecuremode” rejimida faqatgina kommutatorning harakati o’zgaradi va kontrollerga yo’naltirilgan barcha paketlar va xabarlar tashlab yuboriladi. Jadvaldagi qoida “failsecuremode” dagi mos taymaut tugagunicha qolishi kerak. “failstandalonemode” rejimida rezerv qilingan OFPP_NORMAL portini ishlatib, kommutator barcha paketlarga ishlov beradi. Boshqa so’z bilan aytganda kommutator odatdagi Ethernet kommutator yoki marshrutizator sifatida harakat qiladi. Odatda bunday rejim gibrid kommutatorlarda mavjud.

Kontrollerga yana ulanganida mavjud oqimlar yozuvi qoladi. Keyin agar bu kerak bo’lsa, barcha yozuvlarni o’chirish imkoniyatiga ega.

Kommutator birinchi yoqilganida u “failsecuremode” yoki “failstandalonemode” rejimida kontrollerga muvaffaqiyatli ulanmagunicha ishlaydi.

2.3. OpenFlow protokoli

OpenFlow protokoli 3 turdagi xabarlarni qo’llab-quvvatlaydi:

1. Controller-to-switch – kontroller tomonidan ishlab chiqiladi va bevosita kommutator holatini nazorat qilish va boshqarish uchun ishlatiladi.
2. Asinxron xabarlar kommutator tomonidan ishlab chiqiladi va tarmoqdagi hodisalar (xatoliklar, rad etishlar) haqida va kommutator holati o’zgarishi haqida ogohlantirish uchun ishlatiladi.
3. Simmetrik xabarlar kommutator tomonidan ham, kontroller tomonidan ham ishlab chiqilishi mumkin.

Kontroller-kommutator xabarlari. Kontroller tomonidan quyidagi xabarlar ishlab chiqiladi, ular kommutatordan javob talab qilishi yoki talab qilmasligi mumkin:

- Features: kontroller kommutator imkoniyatlarini features so’rovi yordamida so’rashi mumkin; kommutator features yordamida kommutatorning imkoniyati ko’rsatilgan javob qaytarishi kerak. Bu odatda OpenFlow kanal yaratilishida bajariladi.

- Configuration: kontroller kommutatorning parametrlarini o’rnatadi va so’raydi. Kommutator faqat kontroller so’rovsiga javob beradi.

- Modify-State: Modify-State xabarlari kontroller tomonidan kommutator holatini boshqarish uchun jo’natiladi. Ularning bosh maqsadi OpenFlow jadvallarida qo’shishG’o’chirish va o’zgartirish va kommutator portining xarakteristikalarini (parametrlarini) o’rnatish hisoblanadi.

- Read-State: Read-State xabarlari kontroller tomonidan kommutatorlardan statistik ma’lumotlar yig’ish uchun ishlatiladi.

- Packet-out: Packet-out xabarlari kontroller tomonidan kommutatordagi ma’lum portdan paketlarni uzatish va Packet-in xabarlari yordamida olingan paketlarni yo’naltirish uchun ishlatiladi. Packet-out xabarlari butun paketdan yoki kommutatorga yuklangan paketga havola qiluvchi ID buferning identifikatoridan tashkil topishi kerak. Xabar ko’rsatilgan tartibda qo’llanilishi kerak bo’lgan harakatlar ro’yxatidan iborat bo’lishi kerak: agar harakatlar ro’yxati bo’sh, unda paket tashlab yuboriladi.

- Barrier: so’rovG’javob Barrier xabarlari kontroller tomonidan xabarlar o’rtasida bog’liqlik o’rnatishni ta’minlash uchun yoki tugallangan operatsiyalar to’g’risida bildirishnoma olish uchun ishlatiladi. Qachonki xabarlarni ma’lum tartibda ishlov berish zarur bo’lsa ishlatiladi.

- Role-Request: so’rov kommutatorda kontrollerning joriy rolini o’zgartirish (Slave dan Master ga rolini oshirish uchun) uchun ishlatiladi (protokolning 1.3 versiyasi uchun).

- Asynchronous-Configuration: bu xabar kontroller tomonidan kommutatordan olinadigan asinxron xabarlarga filtr o’rnatish uchun ishlatilishi mumkin (protokolning 1.3 versiyasi uchun).

Asinxron xabarlar. Kommutatorlar asinxron xabarlarni kontrollerga paket kelganligi, kommutatorning holatigi o’zgarganligi yoki xatolik to’g’risida bildirish uchun jo’natadi. Asinxron xabarlar quyidagicha turda bo’ladi:

- Packet-in. Kommutatorning jadvalida mos qoida ega bo’lmagan barcha paketlar uchun kommutator Packet-in xabarini generatsiya qiladi va uni kontrollerga uzatadi. CONTROLLER virtual portiga jo’natilgan barcha paketlar uchun, Packet-in xabari kontrollerga uzatiladi. Agar kommutator kontrollerga jo’natilgan paketlar buferida etarlicha xotiraga ega bo’lsa, packet-in xabar paket sarlavhasining qandaydir qismidan (odatda 128 bayt) va qachonki kommutator paketni yo’naltirishga tayyor bo’lganda kontroller tomonidan ishlatiladigan bufer ID identifikatoridan tashkil topadi. Ichki virtuallashtirishni qo’llab-quvvatlamaydigan (yoki ichki xotirani ishlatib bo’lganida) kommutatorlar kontrollerga paketni butunligicha xabarning bir qismi sifatida jo’natishi kerak. Buferlangan paketlar qoidaga ko’ra kontrollerdan Packet-out xabarlar yordamida ishlov beriladi yoki qandaydir vaqtdan so’ng avtomatik o’chiriladi.

- Flow-Removed. Qachonki yangi oqim uchun qoida flow-mod xabarlari yordamida kommutatorga qo’shilsa u uchun taym-aut (qoida turish vaqti) qiymati o’rnatiladi. Bu qoida bu vaqt oralig’idan keyin faollik etishmasligi yoki qoida qo’llanilmaganligi sababli o’chirilishi kerak. Shuningdek, qat’iy taym-aut ko’rsatilishi mumkin, agar yozuv oqimning faolligidan qat’iy nazar o’chirilishi kerak bo’lsa. Flow-mod xabari shuningdek, oqim tugallanganda kommutator kontrollerga oqimni o’chirish haqida xabar jo’natish kerakmi aniqlaydi.

- Port-status. Kommutator port konfiguratsiya holatlari o’zgarishlarida kontrollerga Port-status xabarlarini jo’natish imkoniyatiga ega.

- Error. Kommutator xatolar haqida xabarlar yordamida muammolar haqida kontrollerga ma’lum qilish imkoniyatiga ega.

Simmetrik xabarlar so’rovsiz ixtiyoriy yo’nalishda jo’natiladi. Assinxron xabarlar quyidagicha turda bo’ladi:

- Hello: Kommutator va kontroller ulanish o’rnatilganida Hello xabarlari almashishadi.

- Echo: so’rovG’javob turidagi Echo xabarlar ixtiyoriy kommutator yoki kontroller jo’natishi mumkin, bu holda albatda javob olinishi kerak. Ular kechikishni yoki kontroller-kommutator ulanish o’tkazish qobiliyatini o’lchashda, shuningdek ulanish yashovchanligini tekshirishda ishlatilishi mumkin.

- Experimenter: Experimenter xabarlari OpenFlow xabarlari turlari fazosida eksperimentlar o’tkazish uchun qo’shimcha funktsionallikni ta’minlash uchun mo’ljallangan.

Xabarlarni etkazish:

- OpenFlow protokoli xabarlarni ishonchli etkazishni va ularga ishlov berishni ta’minlaydi, lekin etkazilganligi haqida avtomatik tasdiqlash yoki xabarlarni tartibli ishlov berilishini ta’minlamaydi. Xabarlarga ishlov berish asosiy ulanish va ishonchli ma’lumot uzatishni qo’llanadigan qo’shimcha ulanish uchun ta’minlanadi, lekin ishonchli bo’lmagan ma’lumot uzatishni qo’llanadigan qo’shimcha ulanishlarni qo’llab-quvvatlamaydi (1.3 versiya).

- Xabarlar etkazilib berilishi OpenFlow kanal to’liq rad etgunicha kafolatlanadi, bu vaziyatda kontroller kommutator holati to’g’risida qandaydir taxmin qila olmaydi.

Xabarlarga ishlov berish. Kommutatorlar kontrollerdan olingan har bir xabarga agar bu zarur bo’lsa to’liq javob generatsiya qilish imkoniyati bilan ishlov berishi kerak. Agar kommutator kontrollerdan olingan xabarga to’liq ishlov bera olmasa, u xatolik haqida teskari javob qaytarishi kerak.

Kommutatorlar qo’shimcha kontrollerga kommutator holati o’zgarishi natijasida generatsiya qilingan flow_removed, port_status yoki packet_in xabarlar kabi barcha asinxron xabarlarni jo’natishi mumkin. Bu xabarlar Asynchronous Configuration xabarlari asosida filtrlanishi mumkin. Masalan, kontrollerga jo’natilishi kerak bo’lgan kommutator portida olingan paketlar kommutator yuklanishi sababli yoki kommutator ichki QoS siyosati va packet_in xabarlar generatsiya qilmasligi sababli tashlab yuborilishi mumkin. Bunday paket tashlab yuborishlar yana bo’lishi mumkin, agarda paket jadvaldagi hech bir yozuv bilan mos kelmasa va jadval harakati odat bo’yicha “kontrollerga jo’natish”da turgan

bo’lsa. Kontrollerlar ular olayotgan xabarlarni rad qilishi mumkin, lekin ular kommutator bilan ulanishni yuqotmaslik uchun echo xabarlarga javob berishi kerak.

Xabarlar tartibi. Xabarlar tartibi barrier turidagi xabarlarni foydalanish hisobiga ta’minlanishi mumkin. Bunday xabarlar bo’lmaganida, kommutatorlar unumdorlikni oshirish uchun xabarlarni ixtiyoriy qayta tartiblashtirishi mumkin; shundan keyin kontrollerlar xabarlarga ishlov berish spetsifik tartibiga bog’liq bo’lmasliklari kerak. Hususan, qoida jadvalga kommutatordan olingan flow-mod xabarlardagidan farq qiladigan tartibda qo’yilishi mumkin. Xabarlar barrier xabari orqali qayta tartiblanishi mumkin emas va barrier xabariga faqatgina barcha imtiyozliroq xabarlar ishlov berilib bo’lgandan keyin ishlov berilishi kerak.

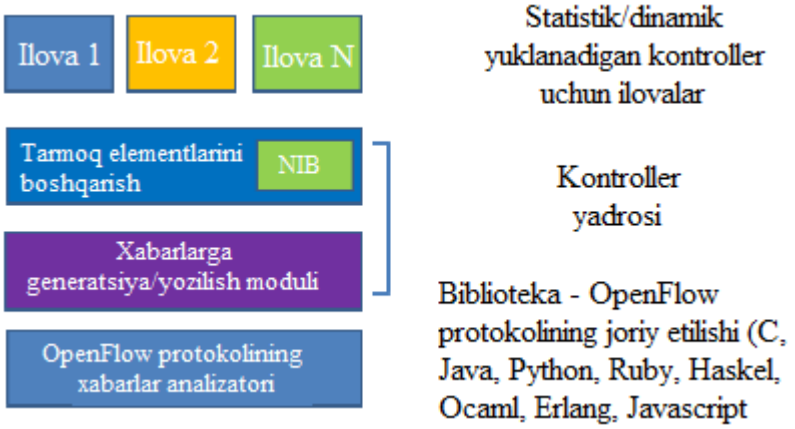
Agar kontrollerdan ikki xabar bir biriga bog’liq bo’lsa (masalan, flow_mod keyingi packet_out bilan OFPP_TABLE ga qo’shiladi), ular ma’lum tartibda buyruqlarga ishlov berish imkonini beradigan barrier xabari bilan ajratilishi kerak.

2.4. Kontroller, tarmoq operatsion tizimlari va tarmoq ilovalari

Dasturiy konfiguratsiyalanadigan tarmoqlarda kalit element kontroller hisoblanadi. Kontroller deyilganda, tarmoq holati va uning elementlarini nazorat qiladigan va boshqaradigan va tarmoqda ma’lumotlar oqimini boshqaradigan, fizik serverga o’rnatilgan maxsus dasturiy ta’minot tushuniladi. Kontroller tarmoq operatsion tizimidan va uning tepasida ishlaydigan tarmoq ilovalari majmuidan tashkil topadi. Tarmoq operatsion tizimi bevosita tarmoq elementlari (kommutatorlar) bilan o’zaro birga ishlashni amalga oshiradi, tarmoq elementlaridan xabarlar asosida tarmoq holatini shakllantiradi, o’zaro ilovalarning birga ishlash imkoniyatini taqdim etadi, tarmoq elementlariga boshqaruvchi harakatlarni tarqatadi ya’ni tarmoqning tarmoq resurslarini boshqarishni amalga oshiradi. Tarmoq ilovalari tarmoq administratori tomonidan yozilgan, tarmoq holati to’g’risida axborotga asoslanib, bevosita trafikni boshqarishni amalga oshiradi (siyosatga, tarmoqning joriy yuklamasi va shunga o’xshashlarga asoslanib). Shunday qilib, kontrollerda kamida bitta ilova o’rnatilishi kerak.

Bugungi kunda barcha kontrollerlar uchun bazaviy sifatida yozilgan bunday ilova learningswitch hisoblanadi.

Kontrollerning umumiy arxitekturasi 9-rasmda keltirilgan.



9-rasm. Kontrollerning umumiy arxitekturasi

Tarmoq operatsion tizimining quyidagi asosiy vazifalarini ajratish mumkin:

- Tarmoq qurilmalarini boshqarish.
- Topologiyani boshqarish (tarmoq topologiyasini qurish, tarmoq yangi elementlarini qo'shishG'o'chirishni amalga oshirish).
- Ilovalarni boshqarish:
  1) Bir vaqtda birnechta ilovalarni yoqish.
  2) Ilovalarni qayd etish.
  3) Ilovalar xatoliklariga ishlov berish.
  4) Ilovalar birga ishlashi xatoliklariga ishlov berish.
  5) Ilovalardan hodisalarga yozilish
  6) Ilovalarga imtiyozlar berish.
  7) Mos infrastruktura yordamida ilovalar o'rtasida o'zaro ishlashni ta'minlash.
  8) Ilovalar yuklamalarini balanslash (oqim bo'yicha).
  9) Tarmoq elementlariga ilovalarning kirish chegaralarini belgilash.
- Serverning mumkin bo'lgan resurslarini boshqarish (oqimlarni, yadrolarni).
  1) Ko'p oqimlilikni qo'llab-quvvatlash (hodisa, ilovalarga ko'p oqimli ishlov berish).

2) Oqimlar yuklanishini monitoring qilish.

- Hodisalar va uzulishlarni boshqarish:
  1) Kommutatorlardan hodisalarga ishlov berish.
  2) Yangi hodisalarni yoqish mexanizmlari.
  3) Ilovalar o’rtasida hodisalarni taqsimlash.

Yangi oqimlar uchun kontroller tomonidan qoidalar ikki rejimda o’rnatilishi mumkin:

- Reaktiv – yangi oqim o’rnatish so’rovnomasiga kontroller bitta qoida shakllantiradi va uni so’rovnoma hosil qilgan kommutatorga o’rnatadi.
- Proaktiv – yangi oqim o’rnatish so’rovnomasiga kontroller oqim uchun marshrut hisoblaydi va bu marshrutdagi barcha kommutatorlarga mos qoidani o’rnatadi.

3. TARMOQ OPERATSION TIZIMLARINING TAHLILI

Mazkur bobning maqsadi dasturiy konfiguratsiyalanadigan tarmoqlarda tarmoq infrastrukturasini va ma’lumotlarni oqimini boshqarish uchun mavjud operatsion tizimlarni (OT) tahlil qilishdan iborat.

Shunday qilib bu bobda mavjud tarmoq OT lari quyidagi parametrlar bo’yicha tahlil qilinadi:

- Umumiy xarakteristikalar;
- Tarmoq OT yaratish maqsadi;
- Joriy etish o’ziga xosliklari;
- Arxitektura va funktsional imkoniyatlari;

Bugungi kunda dasturiy konfiguratsiyalanadigan tarmoqlar uchun quyidagi tarmoq OT lar bizga ma’lum:

- NOX-Classic
- NOX
- POX
- SNAC
- Beacon
- Maestro
- Floodlight
- Trema
- MUL
- ONIX
- Kandoo

3.1. Yaratish maqsadi bo’yicha tarmoq OT larni taqqoslash

3.1-jadvalda tarmoq OT lar DKT uchun yaratilish maqsadi bo’yicha taqqoslanishi keltirilgan.

3.1-jadval. Tarmoq OT larining yaratilish maqsadi

Nomi	DT	Yaratish maqsadi
NOX-Classic	C++, Python	DKT uchun birinchi tajriba kontrolleri. Hozirgi kunda bu versiya rivojlanmayapti. Loyiha ikki mustaqil loyihaga ajratilgan (NOX va POX).
NOX	C++	NOX C++ da yozilgani va ko'poqimli bo'lganligi hisobiga kontrollerning tezkorligini oshirish maqsadida ishlab chiqilgan. NOX Classic dan hosil bo'lgan.
POX	Python	POX OpenFlow texnologiyasi asosida tarmoqlarni boshqarish uchun tezkor ishlab chiqish va nusxalash imkoniyati bilan ilovalarni tadqiq etish va o'rganish uchun ishlab chiqilgan. NOX Classic dan hosil bo'lgan
SNAC	C++	SNAC – Simple Network Access Control – korporativ tarmoqlarda moslashuvchan oddiy siyosatni (kirishni nazorat qilish) aniqlash usullarini ta'minlash uchun ishlab chiqilgan OpenFlow kontroller. SNAC formal modellashtirish tili – formal modelling language (FML) yordamida tarmoqni sozlash imkoniyatini beradi.
Beacon	Java	Java tilidagi DKT uchun birinchi kross-platformali ko'p oqimli tarmoq OT
Maestro	Java	Java tilidagi kross-platformali ko'p oqimli kontroller
FloodLight	Java	Ochiq kodli kontroller. Beacon kontrolleri asosida Java tilida yozilgan. Apache litsenziyasi bo'yicha ishlab chiqilmoqda, ya'ni ixtiyoriy maqsadda foydalanish mumkin.

3.1-jadval davomi.

Nomi	DT	Yaratish maqsadi
Trema	C, Rybu	Trema platforma sifatida C/Rybu tillarida OpenFlow kontrollerlari ishlab chiqish uchun ishlab chiqilgan. Trema tadqiqot va ta’lim muassasalari uchun ishlab chiqilgan, shu sababli o’z ichiga ilovalarni testlash va sozlash muhitlarini olgan.
Mul	C	Mul kontrolleri kritik-muhim masalalarni bajaruvchi tarmoqlarni qurish uchun zarur tezkorlik va ishonchlilikni ta’minlash uchun ishlab chiqilgan.
ONIX	C++, Python, Java	- Boshqarish pog’onasi funktsiyalarini amalga oshirish uchun o’xshash tizimlar bilan taqqoslaganda umumiy API ta’minlash. - ONIX ni turli muhitlarda: WAN, bulut, korporativ ma’lumotlarga ishlov berish markazida foydalanishga yo’naltirilganligi.
Kandoo	C, C++, Python	- Kandoo OpenFlow bilan mos bo’lishi kerak, ya’ni kommutatorlarda ma’lumotlarni uzatish pog’onasida qandaydir yangi funktsionallikni kiritish talab qilinmasligi kerak. - Kandoo boshqaruvchi tarmoq ilovalarini avtomatik taqsimlaydi, ya’ni ilovalar o’zlarining tarmoqda qanday joylashganligi haqida axborotga ega emas va ilova ishlab chiqaruvchilar ularning ilovalari markazlashgan OpenFlow kontrollerda yoqiladi deb faraz qilishadi.

3.2. Umumiy xarakteristikalari bo’yicha OTlarni taqqoslash

3.2, 3.3, 3.4-jadvallarda DKT uchun tarmoq OT lari umumiy xarakteristikalar bo’yicha solishtirma tahlili berilgan. Tarmoq OT larining umumiy

xarakteristikalarining solishtirma tahlili asosida faol rivojlanayotgan va istiqbolli va yashovchan ochiq loyihalarni ajratib olish zarur.

3.2-jadval. NOX-Classic, NOX, SNAC, POX umumiy xarakteristikalarini solishtirish

Xarakte-ristika	NOX-Classic	NOX	SNAC	POX
Ishlab chiquvchi	2008-2009 – Nicira Networks, 2009-2010 – Stanford University, 2009-2010 – ICSI UC Berkeley	2009-2012 – ICSI UC Berkeley (Murphy McCauley, Amin Tootoonchian)	Ed Swierk Rob Vaterlaus (BigSwitch Networks)	James McCauley
Litsenziya	GPL v3	GPL v3	GPL v2	GPL v3
Repozi-toriy	https://github.com/noxrepo/nox-classic.git	https://github.comG’noxrepoG’nox. git	https://github.comG’bigswitchG’s nac-nox	https://github.comG’noxrepoG’pox.gi
Ishlanma boshlanish sanasi	2009 yil 14 mart	2002 yil 11 may	2010 yil 6 may	2012 yil 24 noyabr
Oxirgi versiya sanasi	2010 yil 15 sentyabr	2012 yil 19 dekabr	2011 yil 6 may	2012 yil 24 noyabr
Qo’llab-quvvatlash va rivoj-lanish	Rivojlanmayapti	2012 yil mayda ishlanma tugatilgan	Rivojlan-mayapti	Rivojla-nayapti

3.3-jadval. Beacon, Maestro, FloodLight, Trema umumiy xarakteristikalarini solishtirish

Xarakte-ristika	Beacon	Maestro	FloodLight	Trema
Ishlab chiquvchi	Standford University, David Erickson	Rice University	FloodLight	Trema
Litsenziya	GPL v2 i Stanford University FOSS Licensee Exception v1.0	LGPL v2.1	Apache License v.2.0	GPL v2
Repozi-toriy	git://gitosis.stanford.edu/beacon.git	https://code.google.com/p/maestro-platform/downloads/list	https://github.com/floodlight/floodlight.git	https://github.com/trema/trema.git
Ishlanma boshlanish sanasi	Ma’lumot yo’q	2010 yil dekabr	2011 yil oxiri	Ma’lumot yo’q
Oxirgi versiya sanasi	2012 yil oktyabr	2011 yil may	2013 yil mart	2013 yil mart
Qo’llab-quvvatlash va rivoj-lanish	Rivojlanyapti	Rivojlan-mayapti	Faol rivoj-lanyapti	Faol rivoj-lanyapti

3.4-jadval. Mul, ONIX, Kandoo umumiy xarakteristikalarini solishtirish

Xarakte-ristika	Mul	ONIX	Kandoo
Ishlab chiquvchi	Kulcloud Networks (Dipjyoti Saikia)	Teemu Koponen, Martin Casado,Natashaa Gude, Jeremy Stribling, Leon Poutievski, Min Zhu, Rajiv Ramanathan, Yuichiro Iwata, Hiroaki Inoue, Takayuki Hama, Scott Shenker	Soheil Hassas Yeganeh, Yashar Ganjali, University of Toronto
Litsenziya	GPL v2	Yopiq ishlanma	Yopiq ishlanma
Repozi-toriy	https://git.code.sf.net/p/mul/code mul-code	Yopiq ishlanma	Yopiq ishlanma
Ishlanma boshlanish sanasi	2012 yil 15 avgust	Ma’lumot yo’q	Ma’lumot yo’q
Oxirgi versiya sanasi	2013 yil avgust	2010 yil 4-kvartal	2012 yil 3-kvartal
Qo’llab-quvvatlash va rivoj-lanish	Faol rivoj-lanyapti	Ma’lumot yo’q	Ma’lumot yo’q

Yuqorida 3.2, 3.3, 3.4-jadvallarda berilgan axborotga asoslanib, DKT uchun kontrollerlar ishlab chiqish bo’yicha barcha loyihalarni joriy kunda loyiha qo’llab-

quvvatlanayotganligi bo’yicha ajratish mumkin. Qo’llab-quvvatlashning joriy holati bo’yicha loyihalarni solishtirish 3.5-jadvalda berilgan.

3.5-jadval. Loyihalarning joriy holati bo’yicha taqqoslanishi

Loyihaning joriy holati	Tarmoq operatsion tizimlari
Ma’lumot yo’q	Onix, Kandoo (yopiq ishlanma)
Rivojlanmayapti	NOX Classic, SNAC, Maestro
Rivojlanmoqda	NOX, POX, Beacon
Faol rivojlanmoqda	Mul, Trema, FloodLight

3.3. Joriy etishning o’ziga xosliklari bo’yicha taqqoslash

3.6-jadval. NOX-Classic, NOX, SNAC, POX larni joriy etishning o’ziga xosliklari bo’yicha taqqoslash

Xarakte-ristika	NOX-Classic	NOX	SNAC	POX
Dasturlash tili	C++, Python	C++	C++, Python	Python (2.7)
OpenFlow protokoli versiyasi	OF 1.0	OF 1.0, 1.1, 1.2, 1.3	v0.8.9, v1.0	OF 1.0
Platforma	Linux-platformalar	Ubuntu ning 11.10 va 12.04 versiyalari, Debian va RHEL 6.	Linux-platformalar	Linux, MacOS va Windows Android, BeOS
Ko’p oqimlilik	yo’q	yo’q	yo’q	yo’q
Bazaviy tarmoq OT		NOX-Classic asosida	NOX-Classic asosida	NOX-Classic asosida

3.7-jadval. Beacon, Maestro, FloodLight, Trema larni joriy etishning o’ziga xosliklari bo’yicha taqqoslash

Xarakte-ristika	Beacon	Maestro	FloodLight	Trema
Dasturlash tili	Java	Java	Java	C
OpenFlow protokoli versiyasi	OF 1.0	OF 1.0	OF 1.0	OF 1.0 (OF 1.3 ishlanish jarayonida)
Platforma	Windows, Linux, Mac Android (64 bitli)	Linux	Windows, Linux, Mac Android	Ubuntu12.10, 12.04, 11.10, and 10.04 (i386/amd64, Desktop Edition) Debian GNU/Linux 6.0 (i386/amd64) Fedoraa16 (i386/x86_64)
Ko’p oqimlilik	ha	ha	ha	ha
Bazaviy tarmoq OT			Beacon asosida	

3.8-jadval. Mul, ONIX, Kandoo larni joriy etishning o’ziga xosliklari bo’yicha taqqoslash

Xarakte-ristika	Mul	ONIX	Kandoo
Dasturlash tili	C	C++, Python, Java	C, C++, Python
OpenFlow protokoli versiyasi	OF 1.0 (OF 1.2 rejalashtirilmoqda)	OF 1.0	OF 1.0 (1.1, 1.2 uchun ishlab chiqilmoqda)
Platforma	Ubuntu 10.04 LTS	Linux	Linux
Ko’p oqimlilik	ha	ha	ha
Bazaviy tarmoq OT		NOX-Classic	ONIX root-kontroller sifatida

3.6, 3.7, va 3.8 jadvallarda keltirilganlar asosida barcha tarmoq tizimlari quyidagi mezonlar bo’yicha sinflashtiriladi:

- dasturlash tili bo’yicha taqqoslash natijalari 3.9-jadvalda berilgan;

- OpenFlow protokolining qo’llab-quvvatlovchi versiyalari bo’yicha solishtirish natijalari 3.10-jadvalda berilgan.

3.10-jadval. Dasturlash tili bo’yicha tarmoq OT larni sinflash

Dasturlash tili	Tarmoq OT
C	MUL
Python	POX
C++	NOX
Java	Beacon, Maestro, FloodLight
JavaScript	NodeFlow
C, Ruby	Trema
C++, Python	NOX Classic, SNAC

3.10-jadval. OpenFlow protokolining qo’llab-quvvatlovchi versiyalari bo’yicha tarmoq OT larini sinflash

OpenFlow protokolining versiyasi	Tarmoq operatsion tizimlari
OF 1.0	NOX Classic, NOX, SNAC, POX, Beacon, Maestro, FloodLight, Trema, Mul
OF 1.1	NOX
OF 1.2	NOX
OF 1.3	NOX

3.4. Tarmoq OTlar funtsional imkoniyatlarini va arxitekturasini taqqoslash

NOX tarmoq operatsion tizimi

NOX-Classic – C++ va Python tillarida yozilgan, OpenFlow texnologiyasi asosidagi DKT tarmog’i uchun birinchi kontrollerdir. Hozirgi kunda bu versiya rivojlanmayopti va qo’llab-quvvatlanmayapti. NOX-Classic loyihasi ikki mustaqil ikki mustaqil POX va NOX kontrollerlariga ajratildi. POX – kontrolleri Python da yozilgan, NOX esa tezkorlikni oshirish maqsadida C++ da yozilgan. NOX dasturlashning asinxron hodisali-yo’naltirilgan paradigmaga asosida qurilgan. Bunda kontroller ishlashi kommutatorlardan keladigan xabarlar, ilovalardan keladigan xabarlar, tarmoq OT modullaridan keladigan xabarlar – hodisalar bilan aniqlanadi. Mazkur yondashuv tarmoq OT da hodisalarni tanlash va ularga ishlov berish amalga oshiriladigan kod mavjudligini ko’zda tutadi. Lekin bu yondashuvni amalga oshirish ko’pincha uzoq muddat hodisalarga ishlov berish imkoniyati yo’q, chunki bu vaqtda dastur boshqa hodisalarga javob bera olmaydi. Bu yondashuv serverlarda qo’shiladigan va ko’pincha bir vaqtning o’zida 10 mingdan ortiq ulanishlar kontroller tomonidan qo’llanadigan masshtablashtirish muammolarini

echish uchun server ilovalarida qo’llaniladigan DKT kontrollerlarida foydalanish yaxshi natija bergan.

Arxitektura va asosiy komponentalar. NOX asosiy modullari builtin papkasida joylashgan [8]:

- Kernel moduli – kontroller yadrosi.
- Component moduli – NOX komponentlarini yaratish uchun bazaviy funktsionallikni amalga oshiradi (konfiguratsiyalash, komponentni o’rnatish, hodisaga yozilish, hodisani jo’natish, ishlov beruvchilarni ro’yxatga olish). Komponent NOX tomonidan taqdim etiladigan funktsionallikni inkapsulyatsiyalaydi. Tarmoq ilovasi talab qilingan funktsionallikni ta’minlaydigan o’zaro ishlovchi komponentlarning majmuini o’z ichiga oladi. Har bir komponent har xil holatda (jumladan, o’rnatilmagan, o’rnatilgan, ifodalangan, yuklangan, konfiguratsiyalangan) bo’lishi mumkin. Komponentlar bir birlari bilan statik o’zgaruvchilar bilan o’z aro bog’lanishlari mumkin. Yadro va komponentlar o’zaro ishlashi komponentlar konteksti orqali amalga oshiriladi.
- Connection manager moduli – kommutatorlar bilan ulanish uchun Unix uchun Berkli soketlari ishlatiladi (sysG’socket.h).
- Deployer moduli bo’linuvchi ob’ektlarni yaratish uchun ishlatiladi.
- DSO Deployer (A dynamic shared object deployer) moduli – dinamik bo’linuvchi ob’ektlar yaratish uchun ishlatiladi va harbir komponent uchun kontekst shakllantiradi. DSO – bu ob’ekt fayl bo’lib, ularni bajarish vaqtida NOX komponentlari tomonidan ko’p marotaba umumiy foydalanish mo’ljallangan. NOX dinamik va statik bog’langan komponentlar uchun Deployer lar ko’pligini qo’llab-quvvatlaydi. Deployer kontekst tashkil etadi va uni yadroga uzatadi.
- Static Deployer moduli – statik bo’linuvchi ob’ektlar uchun ishlatiladi.
- Event Dispatcher moduli – hodisalarni boshqarish uchun mo’ljallangan:

1) Ishlov beruvchilar bilan hodisalarni assotsiatsiyalash va mos ishlov beruvchilarga hodisalarni jo’natish. Komponentta amalga oshirilgan: hodisalarni qayd etish, ishlov beruvchilarni hodisalarga qayd etish, har bir hodisa uchun komponentalar ro’yxati bilan mos bu hodisalarga ishlov

berish imtiyozlari. Har bir hodisa uchun ishlov beruvchilar chaqiriqlari zanjiriga mos qo’yilgan.

2) Kutuvchi hodisalar navbatini qo’llab-quvvatlash.

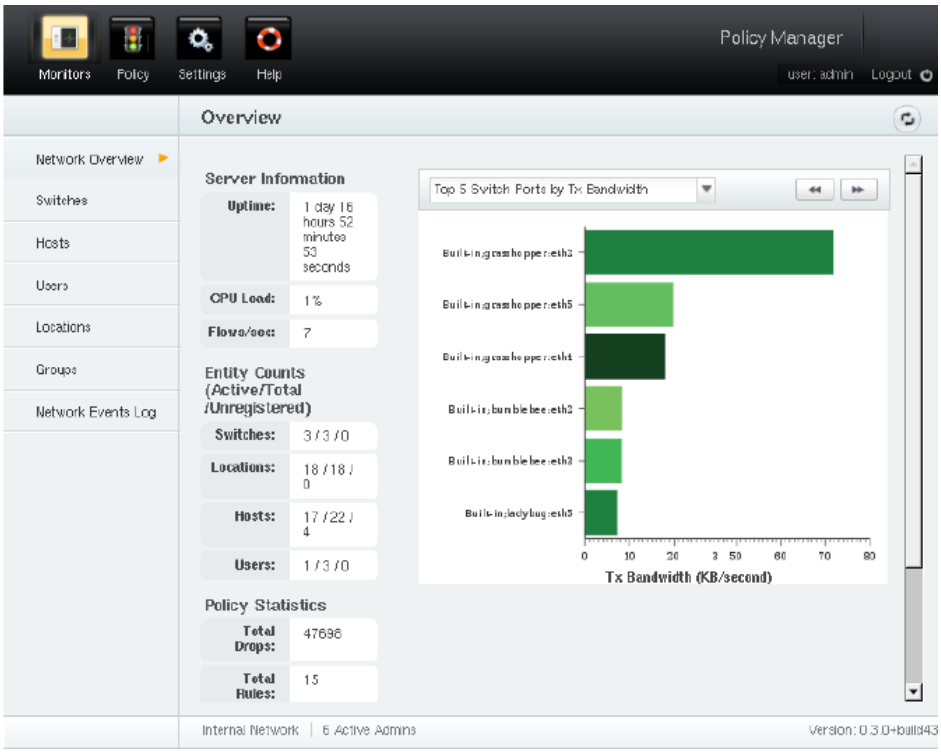
POX tarmoq operatsion tizimi

POX OpenFlow texnologiyasi asosida DKT uchun ilovalarni tezkor ishlab chiqish va nusxalash imkoniyati bilan tadqiq etish va o’rganish uchun yaratildi. Arxiturtura va asosiy komponentlar. POX quyidagi komponentlardan iborat [7]:

1) L2G’L3 forwarding L2G’L3 sathida marshrutlashni ta’minlaydi.
2) Topology discovery – tarmoq asosiy elementlarini topish va tarmoq topologiyasini qurishni ta’minlovchi modul.
3) Topology view – kommutatorlar, xostlar va linklar kabi asosiy ob’ekt modeli hisoblanadigan modul. Bu ob’ektli model boshqa modullar bilan to’ldiriladi.
4) Messenger – tashqi jarayonlar bilan POX uchun interfeys ta’minlovchi komponent.
5) GUI (o’zining va web-interfeys) (POX uchun POX-desk, DjangoFlow web UI)

SNAC tarmoq operatsion tizimi

SNAC – Simple Network Access Control – korporativ tarmoqlar yaratishga yo’naltirilgan OpenFlow kontroller. U NOX 0.4 versiyasiga asoslangan va siyosatni aniqlashning moslashuvchan tili, qurilmalarni sozlash va hodisalarni monitoring qilish uchun foydalanuvchiga qulay interfeysi bilan ajralib turadi. SNAC foydalanuvchi interfeysi 10-rasmda berilgan. SNAC tarmoqni formal modellashtirish tili – formal modelling language (FML) yordamida tarmoqni sozlash imkonini beradi.



10-rasm. SNAC interfeysi

Arxitektura va asosiy komponentlar.

SNAC operatsion tizimi NOX asosida qurilganligi sababli arxitektura va asosiy komponentlar mos keladi. Ayrim yangi komponentlarni va joriy etishning o’ziga xosliklarini keltirib o’tamiz.

1) Packet-Classifier moduli – tushayotgan packet_in xabarlarini ularning mos imtiyozlari bo’yicha klassifikatsiyalaydi.
2) SNAC operatsion tizimi NOX dan farqli haqiqatdan kontroller bilan kommutator o’rtasida SSL asosida himoyalangan ulanishni qo’llab-quvvatlaydi.

Beacon tarmoq operatsion tizimi

Beacon – tezkor, kross-platformali, Java tilidagi modulli kontroller. U ham diskret-hodisali, ham oqimli operatsiyalarni qo’llab-quvvatlaydi.

Arxitektura va uning o’ziga xosliklari.

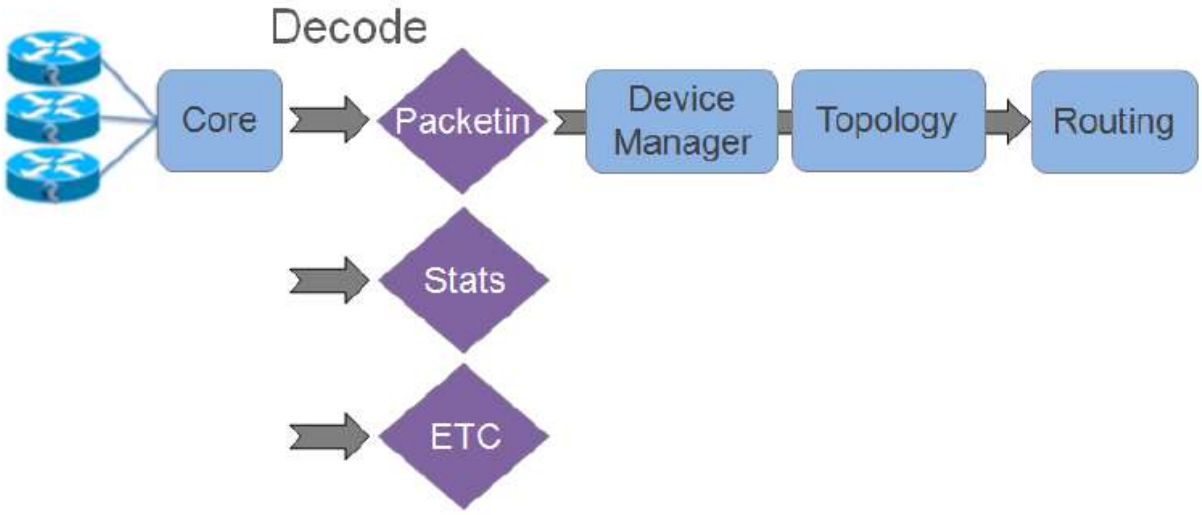
Beacon tarmoq operatsion tizimining asosiy elementi va birligi paket (Bundle) hisoblanadi. Paket o’z ichiga olishi mumkin: metama’lumotlar (META-

INFG’MANIFEST.MA), Java-klasslar, resurslar (xml) va boshqa JAR-fayllar. Paketlarni kontroller ishlash vaqtida, boshqa mustaqli paketlarning bajarilishini to’xtatmasdan qo’shish, to’xtatish, yangilash, o’rnatish mumkin. Shuningdek bir vaqtning o’zida bir paketning birnechta versiyalari mavjud bo’lishi mumkin. Shunday qilib, Beacon kontrolleri o’zida birga ishlaydigan paketlar to’plamini aks ettiradi.

Beacon quyidagi asosiy komponentlardan tashkil topgan [10]:

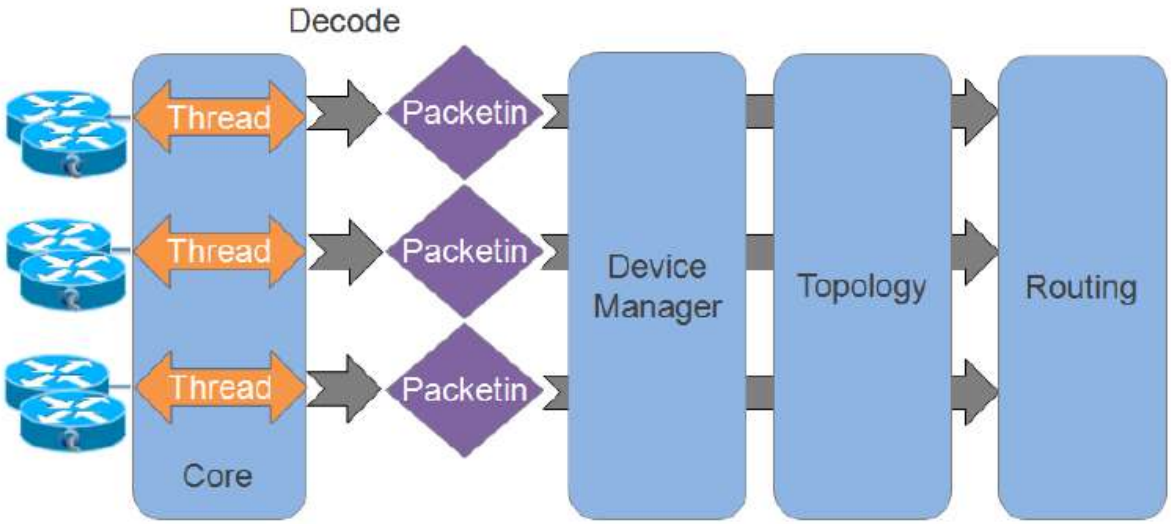
1) OpenFlowJ (OpenFlow v1.0 protokoli) – protokolning Java dagi joriy etilishi;
2) Paketlar shifrator/deshifratori (Ethernet, ARP, IPv4, LLDP, TCP, UDP);
3) Modullar-paketlar: Core, Learning Switch, Hub, Device Manager;
4) Modullar-paketlar: Topology, Layer 2 Shortest Path Routing;
5) ARP Proxy, DHCP Proxy, Multicast eliminator;
6) Deklarativ marshrutlash moduli (matnli faylni yuklash yordamida);
7) Foydalanuvchining veb interfeysi.

Beacon kontrollerining ishlashi quyidagicha qurilgan: yadro paketi (Core Bundle) kommutator bilan ulanadi, kommutatordan xabarlarni olishni amalga oshiradigan va u bilan barcha qolgan paketlar ishlaydigan, IBeaconProvider klassli ob’ekt yaratadi, kiruvchi paketlarga ishlov beruvchi konveyer yaratiladi. 11-rasmda yangi oqim yaratilganligi haqidagi kommutatordan keladigan packet_in xabarlariga ishlov berish amalga oshiriladigan bir oqimli konveyer misol tariqasida keltirilgan. Xabar Core moduliga tushadi, deshifrlanadi, qabul qiluvchini qidirish uchun tarmoq elementlarini boshqaruvchi Device Manager moduliga uzatiladi, keyin qabul qiluvchining tarmoqdagi joyini aniqlash uchun Topology moduliga, yangi ma’lumotlar oqimi uchun marshrut yaratish uchun Routing moduliga uzatiladi.



11-rasm. Paketlarga ishlov beruvchi konveyer

Beacon shuningdek, xabarlarga ishlov berish uchun 12-rasmda ko'rsatilganidek ko'p oqimli konveyerni ham qo'llab-quvvatlaydi.



12-rasm. Paketlarga ishlov beruvchi ko'p oqimli konveyer

Ishonchlilik. Beacon bir yarim yil mobaynida ishlab chiqilgan va bir nechta ilmiy-tadqiqot loyihalarida, tarmoq sinflarida va tajriba segmentlari qo'llanildi. Beacon bugungi kunda 100 ta virtual kommutatorlar bilan ma'lumotlarga ishlov berish tajriba markazining 20 ta fizik kommutatorlar bilan ishlashni qo'llab-

quvvatlaydi va birnechta oy mobaynida to’xtovsiz ishlashni ta’minlash imkoniga ega.

Beacon o’ziga xosliklari:

1) Tezkor ilovalar yaratish imkoniyati – Beacon engil sozlanadi va qo’shiladi. Java va Eclipse tarmoq ilovalarni ishlab chiqish va sozlashni soddalashtirish imkonini beradi.
2) Veb-interfeys.
3) Freymvorklar – Beacon Spring and Equinox kabi Java freymvorklar asosida qurilgan.

Maestro tarmoq operatsion tizimi

Maestro – Rice University universitetida ishlab chiqilgan, Java tilidagi kengayuvchi OpenFlow kontroller bo’lib, tarmoq kirishni va holatini boshqarish modul ilovalarini joriy etish uchun interfeyslar taqdim etadi. Maestro bu modul ilovalar funktsiyalari yordamida tarmoqni avtomatik va dasturiy boshqarishni ta’minlash uchun o’zida platforma aks ettiradi.

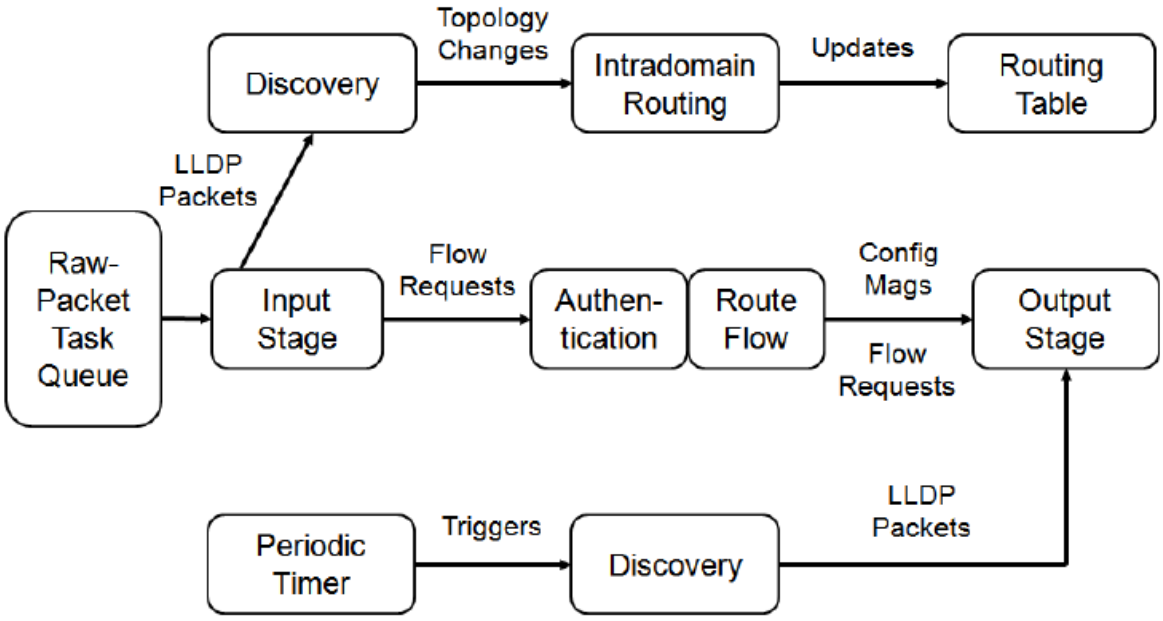
Arxitektura va asosiy komponentlar.

Maestro tizimning o’tkazish qobiliyatini oshirish uchun bir doirasida parallelizmni faol ishlatadi. Dasturiy konfiguratsiyalanadigan tarmoqlarning funktsional o’ziga xosligi shundaki, OpenFlow kontroller oqimning dastlabki faollashishiga javob beradi, shu sababli kontroller tarmoq tor joyi bo’lishi mumkin. Maestro ning afzalligi shundaki, kontrollerda dasturchiga parallellashtirishni tashkil etish bilan bog’liq qiyinchiliklar bartaraf etilgan. Maestro yuklamani taqsimlash masalalarini boshqarish va ishchi oqimlarni rejalashtirish murakkab ishlarning ko’p qismini bajaradi. 13-rasmda Maestro umumiy tuzilishi keltirilgan.

Unumdorlik.

Maestro erishishi mumkin bo’lgan eng katta o’tkazuvchanlik qobiliyati (sekundiga 600000 so’rovnoma) katta ma’lumotlarga ishlov berish markazlariga qo’yiladigan talabdan (sekundiga 20 mln. so’rovnoma) ancha uzoqda bo’lishiga qaramasdan, Maestro NOX da qilinganidek taqsimlanishi kutilmoqda. Buning

natijasida sekundiga 10 mln. masshtabda oqimlarga xizmat ko'rsatishi mumkin bo'ladi. Bundan tashqari, bitta ko'p yadroli serverdagi Maestro ning masshtablanishi, kam deganda NOX da talab qilinadiganidan kontrollerlar sonini 10 marta qisqartirishga yordam beradi.



13-rasm. Maestro umumiy tuzilishi

Floodlight tarmoq operatsion tizimi

Floodlight – kompaniya va korxonalar uchun ochiq kodli OpenFlow Java-kontroller hisoblanadi. Floodlight Beacon kodlaridan paydo bo'lgan. Apache litsenziyasiga ega, ya'ni hohlagan maqsadda foydalanish mumkin.

Toza Java ni ishlatadi (OSGI talab qilmaydi, Eclipse qo'llab-quvvatlaydi, lekin talab qilmaydi). Yig'ish va qo'shish juda sodda.

Big Switch ning Big Network Controller yadrosi hisoblanadi. Floodlight va Big Network Controller uchun ilovalar o'rtasida ko'chib o'tishlikni ta'minlaydi.

Arxitektura va asosiy komponentlar. Floodlight modulli arxitekturaga ega, buning hisobiga kengayish jarayoni va o'zgarish kiritish engillashadi. Arxitekturani ifodalashda ikki asosiy tushuncha ishlatiladi: servis va modul. Servis – bu holatni eksport qiladigan va hodisalar generatsiya qiladigan interfeysdir. Servis iste'molchilari holatni olishlari/o'rnatishlari va hodisalarga yozilishlari yoki

rad etishlari mumkin. Har bir modul o’z navbatida qandaydir funktsionallikni amalga oshirish uchun birnechta servislar to’plamini ishlatishi mumkin. Modul nol yoki birnechta servislarni taqdim etishi mumkin. Ya’ni modullar servislarni eksport qiladi. Floodlight barcha modullari Java da yozilgan. Barcha modullar bir birlari bilan minimal sondagi bog’liqliklarga ega, bu ilovalar ishlab chiqishni osonlashtiradi [9].

O’ziga xosliklari.

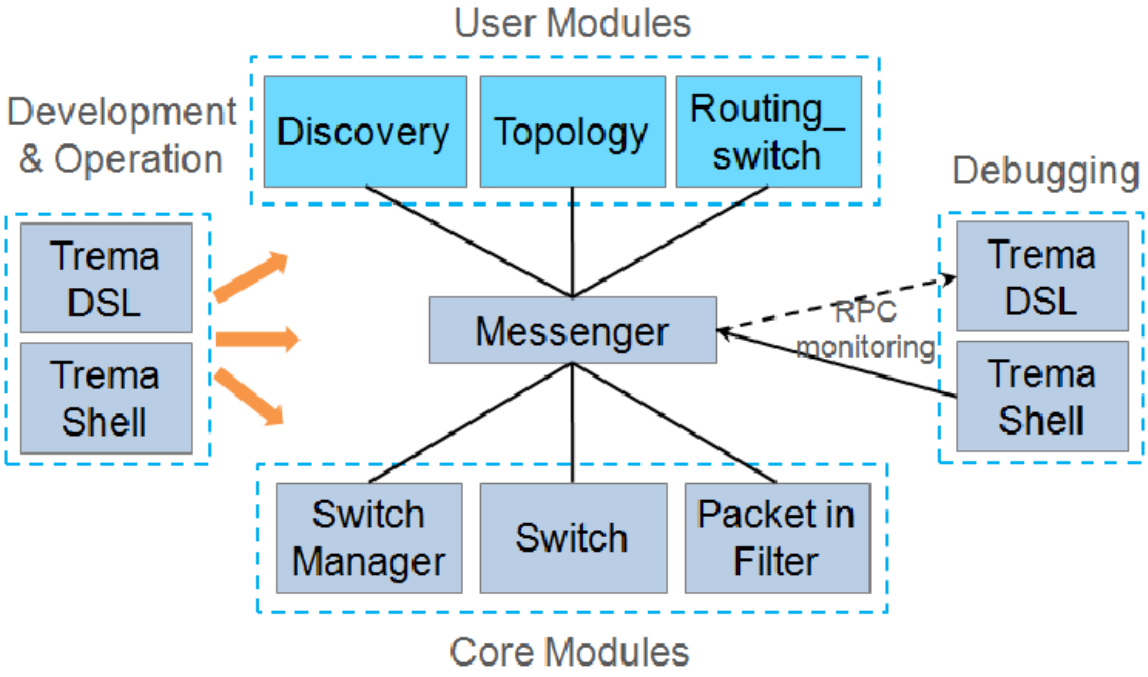
1) Yuklashning modulli tizimi, kontrollerning funktsionalligini takomillashtirish va kengaytirish imkonini beradi.
2) Bog’liqliklarning minimal to’plami bilan engil o’rnatish.
3) Keng diapazondagi virtual va fizik OpenFlow kommutatorlarni qo’llab-quvvatlaydi.
4) OpenFlow-bo’lmagan tarmoqlar bilan integratsiyani qo’llab-quvvatlaydi, ya’ni u fizik OpenFlow kontrollerlarning “orollar” to’plamini boshqarishi mumkin.
5) Ishlanmaning asosiy maqsadlaridan biri – yuqori unumdorlik.
6) OpenStack cloud orchestration platformasini qo’llab-quvvatlaydi.

Trema tarmoq operatsion tizimi

Trema – bu C/Rybu tillarida OpenFlow kontroller ishlab chiqish uchun platforma hisoblanadi. Birinchi ishlar NEC tadqiqotlar laboratoriyasida tezlikni oshirish uchun C, unumdorlikni oshirish uchun Rybu ni foydalangan holda ishlab chiqilgan. Trema platformasi tadqiqot va ta’lim muassasalari uchun ishlab chiqilgan, shu sababli o’z ichiga ilovalarni testlash va sozlash integratsiyalangan muhitini oladi.

Arxitektura va asosiy modullar.

Trema tuzilishi modulli hisoblanadi va 14-rasmda ko’rsatilgan.



14-rasm. Trema tuzilishi

Trema – OpenFlow kontrollerlar yaratish uchun platforma bo'lib, o'z ichiga quyidagilarni oladi:

- Konfiguratsiya komponentlari, bu xabarlarni filtrlash va marshrutlash, virtual tarmoqlarni konfiguratsiyalash, ilovalar va modullarni avtomatik qo'shilishi uchun konfiguratsiyalangan fayl shakllantirish imkonini beradi. Konfiguratsiya Domain Specific Language (DSL) yordamida ifodalanadi.

- Trema yadrosi modullari:

1) Switch Manager komponenti;

2) Switch komponenti;

3) Packet in filter komponenti;

4) Messenger – Trema modullari o'rtasida o'zaro aloqani amalga oshiradi. Bugungi kunda bitta xost doirasida Unix-soketlar asosida Trema modullari o'rtasida point-to-point xabarlar almashish joriy qilingan.

- Qo'shimcha modullar.

- Testlash va sozlash uchun integratsiyalangan muhit:

1) Tarmoq emulyatori;

2) Sozlash vositalari: Tremashark, Wireshark Plugin va loglash vositalari.

Shunday qilib, OpenFlow kontroller – bu Trema va foydalanuvchi modullari (ilovalar) to’plami hisoblanadi.

Ishonchliklik. Trema barqaror va dinamik platforma hisoblanadi, u modullarG’ilovalarning nobarqaror ishlashidan kontrollerni himoyalashni qo’llab-quvvatlaydi: bazaviy funktsionallik hattoki boshqa modullarda rad etayotgan hollarda ham ishlaydi va nobarqaror modullar rad etishlardan so’ng qayta qo’shiladilar.

Mul tarmoq operatsion tizimi.

Mul kontrolleri kritik ahamiyatdagi masalalarni bajaradigan tarmoqlarning unumdorligini va ishonchliligini ta’minlash uchun ishlab chiqilgan (ishonchlilikga yuqori talablar bilan). Mul bulutda ishlatish uchun KulOS tarmoq platformasining (tijoriy kontrollerning) muhim komponenti hisoblanadi va loyiha uning asoschi-kompaniyasi Kulcloud tomonidan qo’llab-quvvatlanib kelinmoqda. Shuningdek, ishlab chiqaruvchilar yana bir maqsad – ilovalarni modulli joylashtirishni qo’llab-quvvatlashni ko’zda tutishdi.

Arxitektura va asosiy komponentlar.

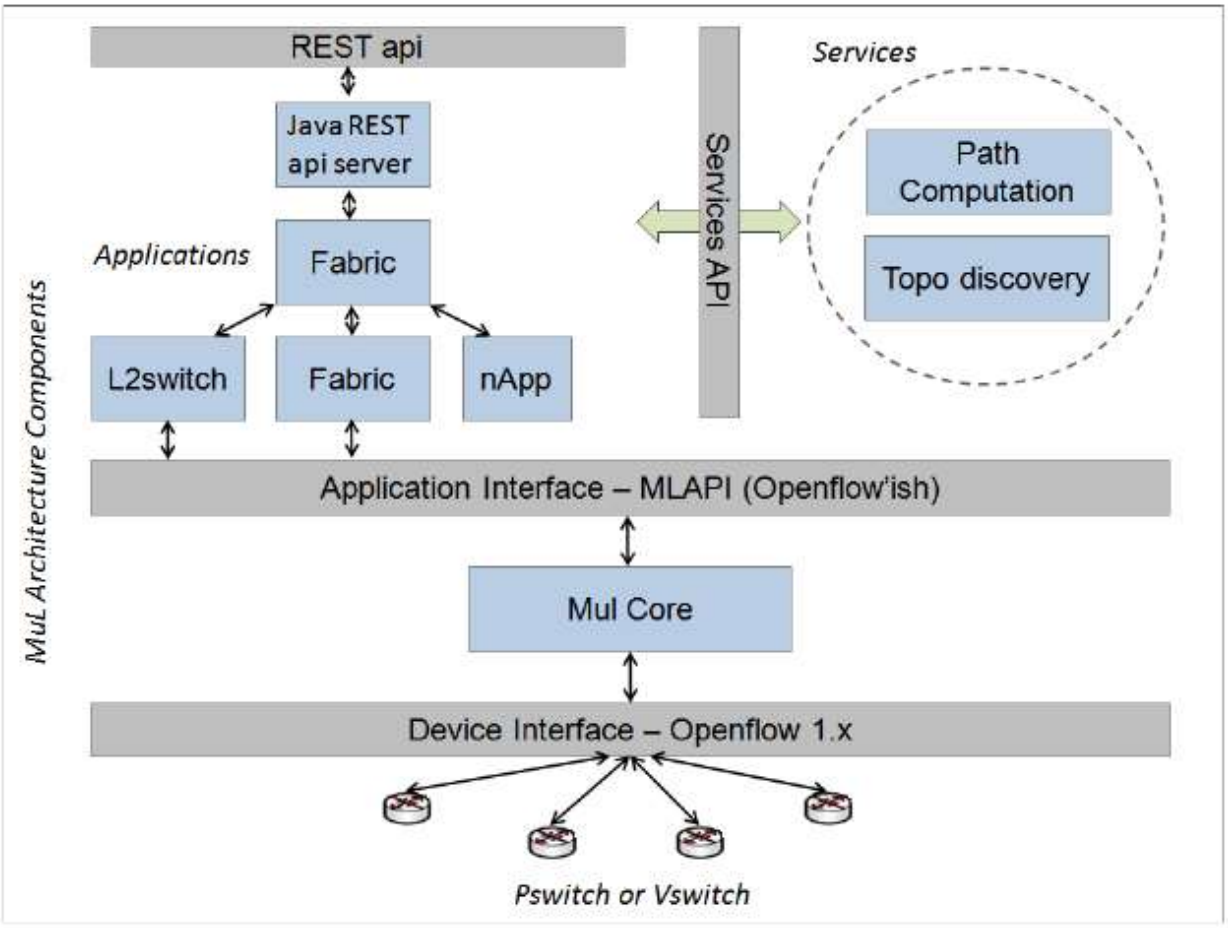
Mul kontrollerining asosiy komponentlari 15-rasmda keltirilgan [5].

Mul kontrollerini o’rnatishning ikki rejimi:

a) odatiy;

b) yuqori unumdorlik bilan.

Kommutatorlar ishlov berish uchun oqimlar soni 0 dan 16 gacha. Ilovalar ishlov berish uchun oqimlar soni 0 dan 8 gacha. Odatda kommutator ishlov berish uchun oqimlar soni – 4, ilovalar ishlov berishi uchun – 2 ni tashkil etadi.

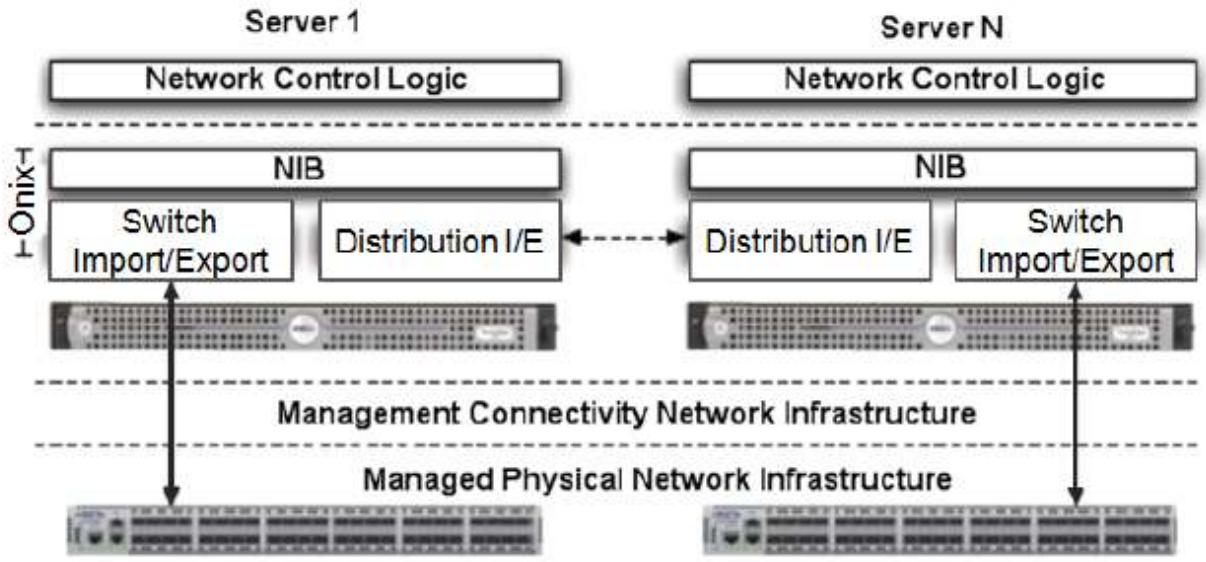


15-rasm. MUL tarmoq operatsion tizimining arxitekturasi

ONIX tarmoq operatsion tizimi.

ONIX – DKT tarmoqlarni boshqarish uchun birinchi taqsimlangan tarmoq operatsion tizimidir. ONIX yopiq ishlanma hisoblanadi. Onix tarmoq operatsion tizimi NOX Classic asosida C++, Python va Java tillarida yozilgan. Taqsimlangan arxitektura bo'lganligi hisobiga u boshqa tarmoq operatsion tizimlarini ishonchlilik va kengayuvchanlik bo'yicha ortda qoldirdi. ONIX turli muhitlarda: korporativ tarmoqlar, ma'lumotlarga ishlov berish markazi tarmoqlari va bulutlarda foydalanish uchun universal platforma sifatida ishlab chiqildi. Lekin murakkabligi va qo'polligi sababli, Onix kengaytirishda qiyinchiliklarga ega. Bugungi kunda bu loyiha faol rivojlanmayapti.

Onix boshqaruvi ostidagi tarmoq arxitekturasi. Arxitekturada Onix bilan boshqariladigan tarmoqning asosiy to'rtta komponenti ajratiladi (16-rasm):



16-rasm. ONIX arxitekturasi

- Fizik infrastruktura: infrastruktura o'z ichiga tarmoq kommutatorlari va marshrutizatorlar, shuningdek hohlagan boshqa tarmoq elementlari (masalan, yuklamani taqsimlagichlar), ular Oniksga tarmoq elementi harakatini nazorat qilgan holda (masalan, forvarding yozuvlari jadvali), holatni o'qish va yozish imkonini beradigan interfeysni qo'llab-quvvatlaydi.

- Ulanish infrastrukturasi: fizik tarmoq qurilmasi va Oniks o'rtasida aloqa ("boshqaruvchi trafik – control traffic) ulanish infrastrukturasi orqali uzatiladi. Bu boshqarish kanali yo ma'lumot uzatish bilan birga (boshqarish trafigi aynan o'sha tarmoqda ma'lumotarni uzatish uchun ishlatiladigan tarmoq forvarding elementlarini (kommutatorlarni) ishlatadi) yo alohida (boshqaruvchi trafikga ishlov berish uchun alohida fizik tarmoq ishlatiladi). Ulanish infrastrukturasi Oniks ekzemplyari va kommutator o'rtasida ikki tomonlama aloqani qo'llab-quvvatlashi va aloqa kanali rad etgan holatda konvergentsiyani amalga oshira olishi kerak. Standart marshrutlash protokollari (masalan, IS-IS yoki OSPF) ulanish infrastrukturasida yo'naltirish holatini (forwarding state) yaratish va qo'llab-quvvatlash uchun mo'ljallangan.

- Oniks: Oniks taqsimlangan tizim hisoblanib, bir yoki bir nechta fizik serverlar klasterlarida ishlaydi, ulardan har biri bir nechta Oniks ekzemplyarlari

bilan ishlashi mumkin. Platforma sifatida Oniks boshqarish tarmoqni boshqarish mantig'iga (tarmoq holatini o'qish va yozish) dasturiy kirishga javob beradi. Juda katta tarmoqgacha (mln. port) kengayish maqsadida, shuningdek ishlab chiqarish tarmoqlarini kengaytirish uchun zarur barqarorlikG'yashovchanlikni ta'minlash uchun Oniks ekzemplyari yana klaster doirasida boshqa ekzemplyarlarga tarmoqning holatini tarqatishga javob beradi.

- Boshqaruvchi mantiq: Tarmoqni boshqaruv mantig'i Oniks API tepasida joriy etiladi. Bu boshqarish mantig'i tarmoqning kutilgan harakatini aniqlaydi; Oniks mos tarmoq holatiga kirish uchun zarur primitivlarni taqdim etadi.

Kandoo tarmoq operatsion tizimi

Kontrollerning kengayuvchanligini ta'minlash uchun serverlarning unumdorligini oshirishdan tashqari ishlov beriladigan OpenFlow xabarlarning sonini qisqartirish hisobiga yuklamani pasaytirishga urinish mumkin. Bu g'oyanining joriy etilishidan biri – tez-tez yuzaga keladigan hodisalarni ularning paydo bo'lish joyida, ya'ni kommutatorlarda ma'lumotlar uzatish sathida ishlov berishdir. [13] ishda ikki sathli arxitekturani ishlatish taklif qilinadi, unda ko'proq tez-tez bo'ladigan lokal hodisalarga ishlov berish quyi sathda (kommutatorlar sathida) amalga oshadi.

Kandoo boshqarish platformasini ishlab chiqishda ikki asosiy maqsad ko'zda tutilgan:

- Kandoo OpenFlow standarti bilan mos bo'lishi kerak, ya'ni kommutator yoki OpenFlow protokoliga qandaydir yangi funktsionallik kiritish kerak emas.
- Kandoo tarmoq ilovalarini administrator ishtirokisiz avtomatik taqsimlaydi, ya'ni Kandoo tarmoq ilovalari ular tarmoqda qanday joylashganligi haqida axborotga ega emas va ilova ishlab chiquvchilar ularning tarmoq ilovalari markazlashgan kontrollerda qo'shiladi deb faraz qilishi mumkin.

Joriy qilishning boshqa o'ziga xosliklari. Kandoo C, C++ va Python tillarida joriy qilingan. Bu C, C++ va Java da yozilgan plaginlarni dinamik yuklash imkonini beradi. RPC API ni qo'llab-quvvatlaydi. Modulli joriy qilingan. Kandoo

OpenFlow protkokolining 1.0 versiyasini qo’llab-quvvatlaydi [6] 1.1 va 1.2 ishlanmoqda).

4. HAYOT FAOLIYATI XAVFSIZLIGI

4.1. Elektr tokidan foydalanishda xavfsizlikni saqlash

Elektr tokidan odam organizmida termik (ya’ni issiqlik), elektrolitik va biologik ta’sir kuzatiladi.

Elektr tokining termik ta’siri inson tanasining ba’zi joyida qo’yish, qon tomirlari, nerv va hujaylarning qizishi sifatida kuzatiladi. Elektrolitik ta’sir esa, qon tarkibidagi yoki hujayralar tarkibidagi tuzlarning parchalanishi natijasida qonning fizik va kimyoviy xususiyatlarining o’zgarishiga olib keladigan holat tushuniladi. Bunda elektr toki markaziy asab tizimi va yurak-qon tizimini kesib o’tmasdan, tananing ba’zi bir qismlarigagina ta’sir ko’rsatishi mumkin.

Mahalliy elektr ta’siri natijasida kuyib qolish, elektr belgilari hosil bo’lishi, terining netallashib qolishi hollarini ko’rsatishi mumkin. Elektr ta’siridan kuyish, asosan organizm bilan elektr o’tkazgichi o’rtasida volta yoki hosil bo’lganda sodir bo’ladi. Elektr o’tkazgichdagi kuchlanishning ta’siriga qarab bunday kuyish turlicha bo’ladi. Engil kuyish faqat yallig’lanish bilan chegaralanadi, o’rta og’irlikdagi kuyishda pufakchalar hosil bo’ladi va og’ir kuyishda hujayra va terilar ko’mirga aylanib, og’ir asoratlarga olib kelishi mumkin. Elektr belgilari - bu terining ustki qismida aniq kulrang och-sarg’ish rangli 1-5 mm diametrdagi belgi pay do bo’lishi bilan ifodalanadi. Bunday belgilar odatda xavfli emas. Terining metallashib qolishida, odatda, erib may da zarrachalarga parchalanib ketgan metall teri ichiga kirib qoladi. Bu holat ham elektr yoyi hosil bo’lganda ro’y beradi. Ma’lum vaqt o’tgandan keyin bu teri ko’chib tushib ketadi va hech qanday asorat qoldirmaydi.

Agar tok ta’sirida hushini yo’qotgan, ammo nafas olishi va yurak tizimi ishlayotgan bo’lsa, uni quruq va qulay joyga yotqizish, kamari va yoqasini bo’shatish va sof havo kelinishini ta’minlash zarur. Nasharit spirti hidlatish, yuziga suv purkash, tanasini va qo’llarini ishqalash yaxshi natija beradi.

Doimiy elektr tokida, shuningdek sanoat chastotasidagi (50 Gts) o’zgaruvchan toklarda bir xil muhitda tarqalayotgan elektr tokining statsionar elektr maydoni bor deb qarash mumkin. Uni bu maydonning kuchlanishi E(V/m) tok zichligi 8(Ag’m) bilan 8qE/p nisbatan bog’langan va bu Om qonunining differensial formada ko’rinishini tashkil etadi. Bunga asoslanib shu maydondagi xohlagan nuqtani, masalan, A nuqtaning potensialini aniqlash oson.

Elektr tokidan jarohatlanishni oldini olishga qaratilgan asosiy chora-tadbirlar quyidagilardir:

kuchlanish ostada bo’lgan o’tkazgichlarni qo’l yetmaydigan qilib bajarish;

elektr tarmoqlarini ayrimjoylashtirish;

elektr qurilmalari korpusida elektr tokining hosil bo’lishiga qarshi chora-tadbirlar belgilash:

kam kuchlanishga ega bo’lgan elektr manbalaridan foydalanish;

2 qavatli muhofaza qobiqlari bilan ta’minlash;

potensiallarni tekshirish;

erga ulab muhofazalash;

nol simiga ulab muhofazalash;

muhofaza o’chirish qurilmalari;

maxsus elektr muhofazalash;

elektr qurilmalarini xavfsiz ishlatish tashkiliy chora-tadbirlarini qo’llash. Xavfli xonalar: bularga nisbiy namligi uzoq vaqt 75% va undan yuqori bo’lgan nam xonalar, uzoq vaqt havo harorati 30^{o}C dan yuqori bo’lgan issiq xonalar, xona havosida tok o’tkazuvchi changlar - ko’mir va metallarning changlari tok o’tkazgichlar va elektr qurilmalari ichiga kirib elektr xavfi vujudga keladigan, tok o’tkazuvchi polga (metall, temir-beton, er, g’ishtli pollar) ega bo’lgan va ishlayotgan ishchi bir vaqtning o’zida bir tomondan yerga ulangan metall konstruksiyalar, texnologik jihozlarga hamda 2-tomondan elektr qurilmalarining metall korpuslariga tegib ketishi mumkin bo’lgan sanoat xonalari kiradi.

Bunga mashinasozlik sanoati korxonalari sexlari, zinapoyalar, isitilmaydigan omborlar va boshqalarai misol keltirish mumkin.

O’ta xavfli xonalar: bularga namligi juda yuqori bo’lgan (100%, devor, shift, pollarda suv tomchilari hosil bo’ladi), harorati 35°C va undan yuqori, havo tarkibida kimyoviy aktiv moddalar bo’lib, bular elektr o’tkazgichlarning muhofaza qobiqlarini yemirish xususiyatiga ega bo’lgan, shuningdek, xavfli xonalarga xos bo’lgan belgilari mavjud sanoat korxonalarining ish bajaradigan joylarini kiritish mumkin.

Sun’iy erga ulash qurilmalarining yotiq va tik holarda o’rnatilgan metall tayoqchalardan tashkil topgan turlari bo’ladi. Erga ulash qurilmasining vertikal o’rnatiladigan turi uchun diametri 3 - 5 sm bo’lgan po’lat trubalar va 40x40 va 60x60 mm li po’lat uchburchaklarning 2.5 - 3 m uzunlikdagi bo’laklari olinadi. Ularni 0.5m chuqurlikdagi ariqchalar qazilib, ma’lum oraliqda yerga qoqib chiqiladi va o’zaro po’lat polosa yordamida payvandlab biriktiriladi. Po’lat polosa qirqim yuzasi 4x12 mm dan kam bo’lmasligi kerak. Polosa o’rniga diametri 6 mm dan kam bo’lmagan dumaloq po’lat tayoqchalardan foydalanish mumkin. Tabiiy erga ulash qurilmalari sifatida, suv yoki boshqa narsalar uchun yerga o’rnatilgan truboprovodlardan (bunda portlovchi va yengil yengil alangalovchi suyuqliklar va gazlar uchun o’rnatilgan truboprovodlardan tashqari artezan quduqlari truboprovodlari, yerga ulangan qismlarga ega bo’lgan binolarning temir-beton qismlari, kabellarning qo’rg’oshin qobiqlari va boshqalar) dan foydalanish mumkin.

Nolga ulab muhofaza qilishning vazifasi yerga ulab muhofaza qilishniki bilan bir xil, ya’ni elektr asbobi korpusiga oqib ketgan kuchlanishni zararsizlantirishdan iborat. Nolga ulab muhofaza qilishning ishlash uslubi shundayki, kofusga o’tib ketgan elektr tokini nol simi bilan ulash hisobiga qisqa to’qinish vujudga keltirish bilan, elektr qurilmasiga kelayotgan tok kuchining ortib ketishiga erishiladi va buning natijasida elektr qurilmasini muhofaza qilish uchun o’rnatilgan saqlovchi eruvchi qurilmani yoki saqlovchi avtomatni o’chirish bilan elektr qurilmasiga kelayotga elektr toki uzib qo’yiladi. Bunday vazifani bajaradigan saqlovchi eruvchi qurilmalar yoki avtomatlar oldindan elektr qurilmasidagi elektr tokining ma’lum miqdorda oshishiga mo’ljallab qo’yiladi.

Muhofazalovchi o’chirish qurilmasi xavf hosil qilingan elektr asbobni 0.2 s dan oshmagan vaqt davomida o’chirish imkoniyatini berishi kerak.

Muhofazalovchi o’chirish asbobi bir qancha qismlardan tashkil topgan bo’lib, asosan elektr tizimida biror bir parametrning o’zgarishini sezih, elektr tizimiga berilayotgan tokni avtomatik uzuvchi qurilmaga signal beradi. Bu elementlarning asosiysi qabul qiluvchi qurilma bo’lib (asosan qabul qiluvchi qurilma sifatida rele qo’llaniladi), i elektr tizimidagi parametr o’zgarishini qabul qiladi, agar kelayotgan signal kuchsiz bo’lganda, uni kuchaytiruvchi qurilma o’rnatiladi, shuningdek bu tizimning to’g’ri ishlayotganini tekshirib turuvchi nazorat asboblari hamda signal lampalari o’rnatilishi mumkin.

Agar tok ta’sirida hushini yo’qotgan, ammo nafas olishi va yurak tizimi ishlayotgan bo’lsa, uni quruq va qulay joyga yotqizish, kamari va yoqasini bo’shatish va sof havo kelinishi ta’minlash zarur.

Elektr toki ta’siriga tushgan kishiga tibbiyot xodimi kelgunga qadar ko’rsatiladigan yordamni 2 qismga bo’lib qaraladi:

1) tok ta’siridan qutqarish; 2) 1-yordam ko’rsatish.

Agar buning iloji bo’lmasa (masalan, o’chirish qurilmasi uzoqda bo’lsa), unda tok kuchlanishi 1000 V dan ortiq bo’lsa, unda dielektrik qo’lqop va elektr izolyatsiyasi mustahkam bo’lgan elektr asboblaridan foydalanish kerak.

Agar elektr kuchlanishi ostida bo’lgan elektr o’tkazgichning bir uchi erga tegib tursa, unda elektr toki erga oqib" o’ta boshlaydi. Bunday holat tasodifiy maqsadli bo’lishi mumkin. Maqsadli bo’lgan tokning oqib o’tishini erga ulash yoki elektrod deb ataladi.

4.2. Yong’in xafvsizligi qoidalari

Yonish jarayonini shartli ravishda quyidagi turlarga bo’lish mumkin:

chaqnash-yonuvchi aralashmaning bir lahzada yonib-o’chishi. Bunda yonishning davom etishi uchun aralashma tayyorlashning hojati yo’q;

natijada yonishning vujudga kelishi;

alangalanish-yonishning alanga olib davom etishi;

o’z-o’zidan yonish - moddalar ichida asosan organik moddalarda ro’y beradigan ekzotermik reaksiyalar natijasida, tashqaridan qizdirishsiz yonuvchi aralashmaning o’z-o’zidan yonib ketishi;

o’z-o’zidan alangalanish - o’z-o’zidan yonishning alanga bilan davom etishi;

portlash - o’ta tez yonish jarayoning bosim va energiya hosil qilish bilan o’tishi.

Yonuvchi modda ma’lum haroratlarda o’zidan ma’lum bug’lar ajratib chiqarishi natijasida muhim alangalanish ta’minlansa, bu harorat alangalanish harorati deb yuritiladi. Mehnatni muhofaza qilish - bu iqtisodiy, ijtimoiy, texnika, sanitariya-gigiyena, mehnat qonunlari va tashkiliy chora-tadbirlar tizimidan iborat bo’lib, uzluksiz faoliyat jarayonida inson sog’lig’i va mehnat qobiliyatini ta’minlashga qaratilgan fandir.

Insonning jamiyatni taraqqiy etirish hamda ishlab chiqarishni boshqarishda asosiy kuch ekanligini hisobga olib, uning xavfsizligini va sog’lig’ini saqlash ijtimoiy taraqqiy ot yo’lidagi muhim omil hisoblanadi. Shuning uchun ham sanoat korxonalarida mahsulot yetishtirish jarayonida ishlab chiqarish sharoitini yaxshilash, ishlab chiqarishda jarohatlanish va kasb kasalliklarining kelib chiqish manbalarini yo’qotish, shuningdek ish faoliyati inson uchun charchash, toliqish va kasallanish manbai bo’lmasdan, quvonch va baxt keltiruvchi faoliyat bo’lishini ta’minlashga harakat qilish zarur. Sanoat korxonalarida to’kis sanitariya-gigiena sharoitlarini ta’minlash, og’ir qo’l kuchi bilan bajariladigan mehnatni tugatish va kasb kasalliklarini butunlay yo’qotish chora-tadbirlarini amalga oshirish kerak, zero, mehnat qilish faqat yashash vositasi bo’lib qolmasdan, balki hayot talabi bo’lib qolishi kerak.

Xulosa

Mavjud tarmoq operatsion tizimlarini qilgan tahlilimiz bo’yicha quyidagicha xulosalar qilish mumkin:

- Ko’pchilik mavjud tarmoq operatsion tizimlar C, C++, Python, Java, Rybu kabi yuqori darajali tillarda yozilgan.

- Faqatgina Floodlight, Beacon va Maestro Java-kontrollerlari kross platformali hisoblanadi, qolganlari Linux-platformalarda ishlaydi.

- Ko’rilganlardan POX dan tashqari hammasi ko’p oqimli tarmoq operatsion tizimi hisoblanadi.

- Barcha tarmoq operatsion tizimlar OpenFlow protokolining 1.0 versiyasini qo’llab-quvvatlaydi va shundan kelib chiqib, barcha mavjud dasturiy va apparat kommutatorlar bilan ishlaydi.

- Barcha kontrollerlar turli maqsadlarda ishlab chiqilgan, shu sababli turli muhitlardagi (WAN, ma’lumotlarga ishlov berish markazlari) barcha masalalarni echadigan universal kontroller yo’q. Harbir kontroller o’z afzalliklari va kamchiliklariga ega. Masalan, POX boshqalariga nisbatan uning uchun tezda ilova yaratish imkonini beradi. Lekin, C, C++, Java – kontrollerlar ilova yozishda ko’p ish talab qilsa da, tezkorlik bo’yicha etakchi hisoblanadi. Taqsimlangan kontrollerlar murakkabroq arxitekturaga ega va murakkabroq algoritmlarni talab qiladi, lekin unumdorlik, kengayuvchanlik va boshqarish sathining ishonchliligini oshirish imkonini beradi.

- Arxitektura nuqtai nazaridan barcha kontrollerlar modulli tamoyilni ishlatadi, bu kontrollerning funktsional imkoniyatlarini kengaytirgan holda, tarmoq OT ichki ilovalari uchun tarmoq servislarini kengaytirish imkoniyatini beradi. Barcha kontrollerlar juda o’xshash bazaviy servislar to’plamiga ega: OpenFlow xabarlariga ishlov beruvchi servislar, kommutatorlar bilan ulanishni qo’llab-quvvatlash, modullar va ilovalar uchun hodisalar yozilishG’generatsiya qilish mexanizmlari, tarmoq qurilmalarini boshqarish va tarmoq topologiyasini qurish mexanizmlari.

- Ishonchlilik nuqtai nazaridan mavjud kontrollerlar kontrollerlar va ularning ilovalari rad etishi bilan bog’liq tahdidlarga qarshi tura olmaydi, chunki ularda bunday mexanizm kiritilmagan. Taqsimlangan kontrollerlar bundan mustasno: Onix rad etgan kontroller tugunidan soz kontrollerga boshqarishni uzatish qobiliyatiga ega, Kandoo esa lokal kontroller rad etganida boshqaruvni yuqori sathga – tayanch kontrollerga uzatadi.

O’tkazilgan tahlil natijasida Beacon, Floodlight (Java tilida) Trema va Mul (C tilida) ko’proq istiqbolli va faol rivojlanayotgan tarmoq operatsion tizimlari ekanligini aytishimiz mumkin.

Foydalanilgan adabiyotlar