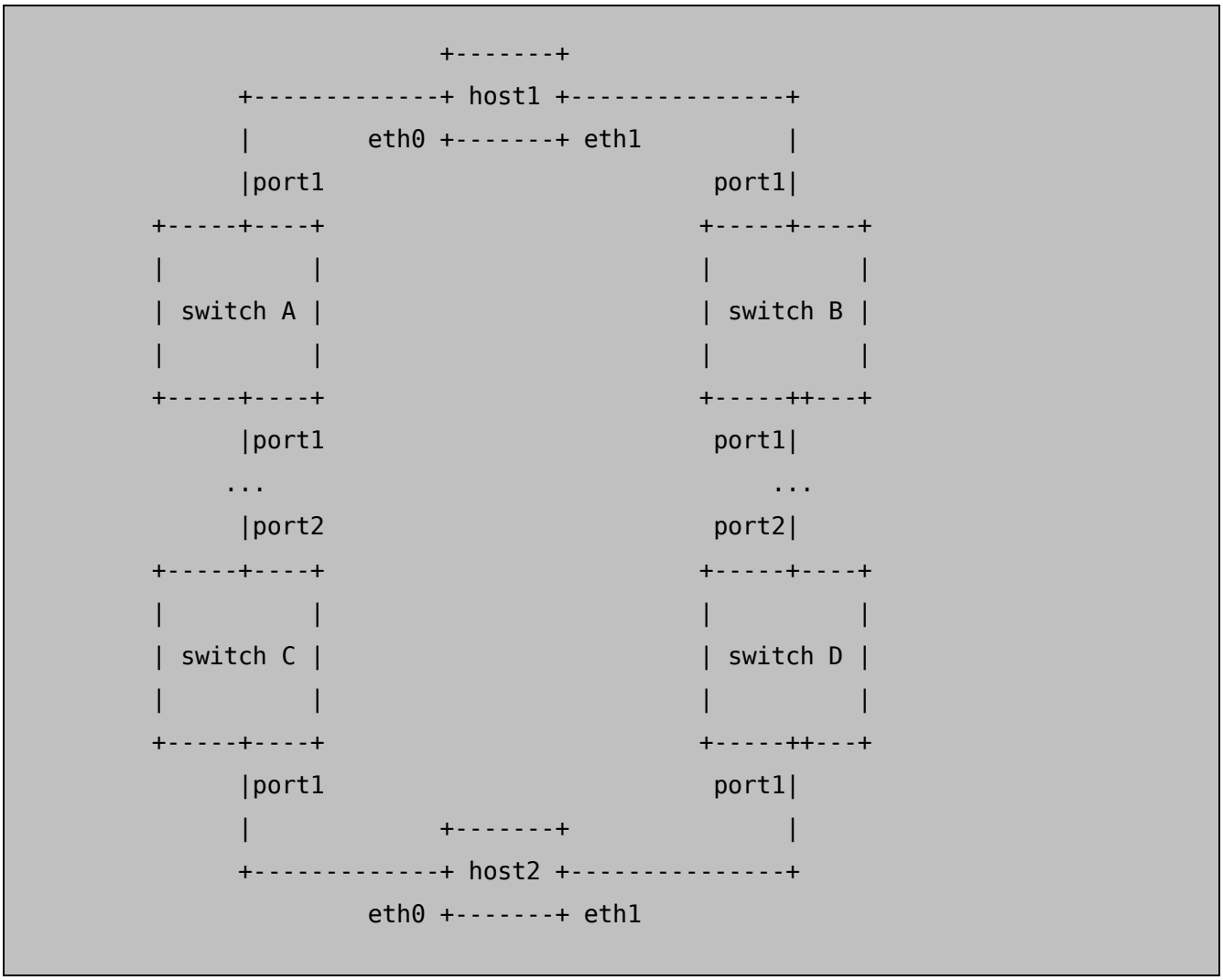
High Availability IP

Karel Rank

Zadání

Je dáno koncové zařízení(KZ1), které má dvě a více rozhraní(I). Každé rozhraní má jinou IP adresu. Komunikace probíhá na úrovni protokolů IP.



Požadavky

V případě výpadku linky na libovolném rozhraní, nesmí dojít k přerušení sezení mezi komunikujícími zařízeními. Tedy druhé koncové zařízení(KZ2), „nesmí“ poznat, že KZ1 má problémy s rozhraním nebo s konektivitou.

Popis situace

Jedná se o propojení point-to-point. Komunikace může probíhat pomocí protokolu TCP(RFC 793) nebo UDP(RFC 768).

Řešení pro HA jsou clustery nebo svázaní rozhraní do virtuálního. Pro clusterové řešení jsem uvedl pouze jeden zajímavý protokol Trickles. Ostatní řešení jsou agregační a jsou popsána v jednotlivých operačních systémech.

Nalezená řešení

Trickles

V článku je popsán bezstavový protokol i s implementací bezstavového stacku, který umožňuje migraci spojení na jiný server. Zajištění migrace je zajištěno díky tomu, že v paketech je zaznamenán aktuální stav serveru. Podle mého názoru toto neřeší zadání a to z několika důvodů:

- implementace vlastního stacku a z toho vyplývající problémy(udržování kódu, další protokol)
- možnost DDoS, i když jsou v článku popsány způsoby, jak mu předejít
- o hodně věcí se musí starat klient
- je to řešeno pro více serverů

Linux Bonding Driver

Jedná se o driver v Linuxovém jádře. Umožňuje „svázat“ několik rozhraní(sítových karet) do jednoho virtuálního.

Původně pochází z patchů pro jádra 2.0 z projektu boewulf. Je navržen pro HA služby. Má několik režimů a v podstatě každý podporuje dva typy: load balancing nebo hot standby, případně kombinace obou dvou. Níže jsou uvedeny jen ty režimy, které zajišťují fault tolerance.

balance-rr Jednotlivá rozhraní se postupně vysílají pakety(proto

RoundRobin). Tento režim zajišťuje loadbalancing a fault tolerance.

active-backup Je vysíláno pouze přes jedno rozhraní. Pokud dojde k jeho selhání, je vysíláno přes další volné. Je zajištěna fault tolerance.

balance-xor Vysílací rozhraní je vybráno na základě hashovací metody. Lze vybrat z různých hashovacích technik. zajišťuje load balancing a fault tolerance.

broadcast Na všechny rozhraní jsou pakety posílány zároven. Zajišťuje fault tolerance.

802.3ad Dynamická agregace linek podle IEEE 802.3ad. Ne všechny režimy.

active-backup nepotřebuje speciální nastavení portů na switchy. Ostatní uvedené režimy potřebují mít porty na switchy nastaveny do „etherchannel“(dle cisco terminologie).

Detekce výpadku linky se může provádět pomocí 2 mechanismů. Přes MII nebo ARP. MII detekuje pouze stav zapojení rozhraní, tzn. jestli je kabel připojený a podle dokumentace může zjištění výpadku trvat delší dobu, než je žádoucí. ARP mechanismus je propracovanější. Posílá ARP dotazy jednomu nebo více zařízení na síti. Adresy těchto zařízení jsou konfigurovatelné. Bohužel řeší výpadky v rámci vnitřní sítě(ne u providera, případně na cestě ke klientovi). Oba mechanismy nelze (z implementačních důvodů) používat zároveň.

Při nasazování jsem se setkal s problémy, pokud svázané zařízení nejsou stejného typu(chipsetu).

Výhody:

- řeší výpadky linky v podstatě pod úrovní IP stacku, běžící aplikace nemusí mít pojem o topologii sítě
- řešení běžně používané v produkčním prostředí
- je zadarmo

Nevýhody:

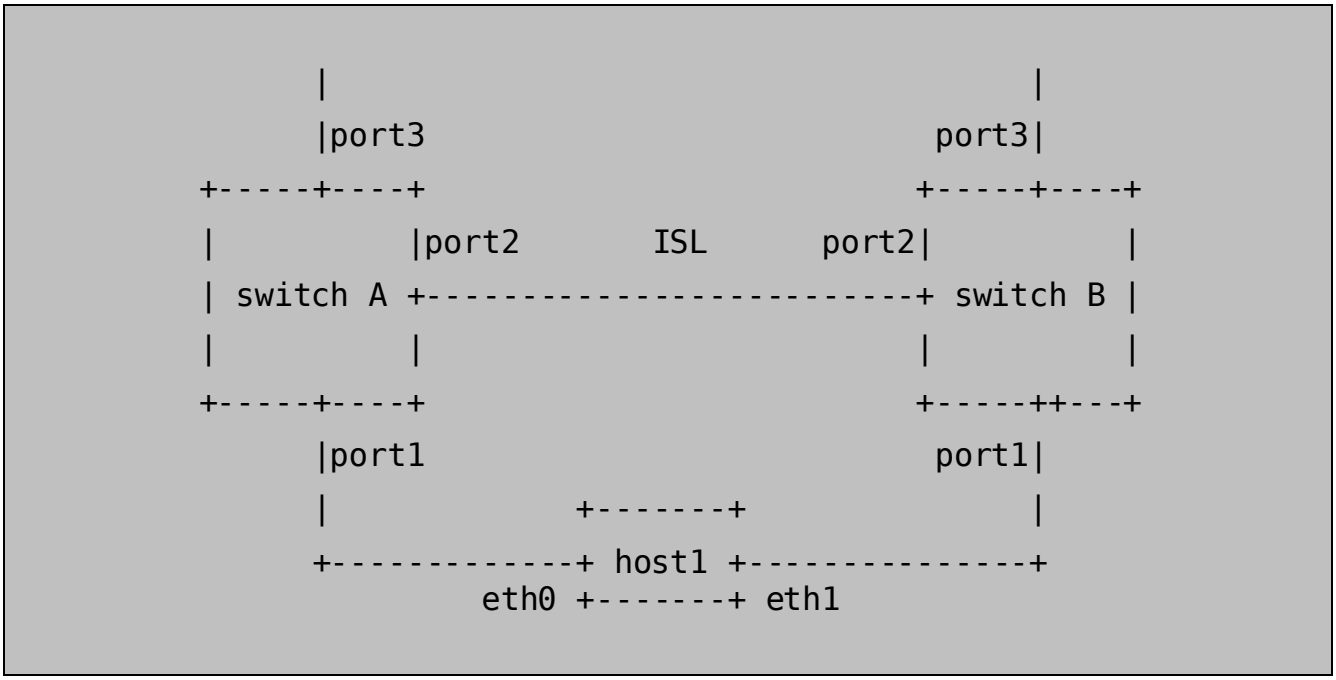
- nedetekuje výpadky linky mimo vnitřní síť, tedy na cestě ke klientovi
- pouze pro Linux
- licence GPL, takže při případné modifikaci se musí zveřejnit kód

Linux Bonding Driver doplněný o link-state routing protocol

Protože samotný bonding driver neřeší problémy na lince, měl by být rozšířen o routovací protokol s detekcí stavu linky.

Možnosti tohoto řešení jsou 2. Modifikovat bonding driver, aby místo ARP detekce používal OSPF, případně kombinaci obou. Druhé řešení je zapojit switche a routery do failover řešení, tzn. že bonding driver se nemodifikuje a detekce výpadků linky je prováděna samotnými routery.

Zde je ukázka možného zapojení switchů se serverem ve failover zapojení[1].



Switch A a switch B jsou propojeni pomocí Inter Switch Link protokolu.

AIX, EtherChannel a 802.3ad

V operačním systému AIX je implementován EhtherChannel nebo 802.3ad.

1 je použito z dokumentace pro bonding driver

EtherChannel nemá nic společného s Cisco. Jedná se o svazování rozhraní do jednoho virtuálního.

EtherChannel vyžaduje konfiguraci switche a umožňuje různé strategie odesílání paketů(výběr podle algoritmu nebo round-robin). Jeho chování je stejné jako u Bonding driveru.

802.3ad nepotřebuje žádnou konfiguraci switchi(narozdíl od EtherChannel). Podporuje pouze standardní distribuci paketů.

IPMP

IP network multipathing je v Solarisu. Umožňuje, obdobně jako bonding driver, svázání fyzických rozhraní do jednoho virtuálního.

Jedno a více zařízení se dají do IPMP skupiny. Pokud jsou v ní aspoň dvě zařízení, je tato skupina failover. Do skupiny se dají přidat pouze rozhraní se stejným typem média(nelze dát např. ATM a ethernet). Na druhou stranu lze přidat do skupiny rozhraní různých rychlostí. Každé rozhraní ve skupině může mít svou vlastní adresu nebo skupina má jednu jedinou adresu. Pokud dojde k výpadku rozhraní s vlastní adresou, je adresa přenesena na fungující rozhraní.

Detekce chyb je řešena jednak aktuálním stavem portu na switchy a odesíláním a zpětným příjmem ICMP paketů. Při vysílání ICMP paketů dochází dost často ke zpožděné odezvě od cílů, protože routery jsou jako ochranu před DoS útoky mají nastavenou nejnižší prioritu pro ICMP.

V současnosti je tato služba dostupná jak v komerční verzi Solarisu, tak i v OpenSolaris. V OpenSolaris dochází k jeho opětovnému návrhu.

one2many

Agregace a failover interface z FreeBSD 4.2. Umí posílat pakety na všechny rozhraní nebo používat round-robin. Nedosahuje takových kvalit jako dříve uvedená řešení, protože neumožňuje lepší detekci výpadků než jen pomocí stavu zapojení rozhraní.

Závěr

Nejvíce vyhovující řešení je Bonding driver doplněný o routovací protokol. Neměl jsem bohužel možnost toto zapojení ověřit v praxi. Bonding driver běžně používáme v mém zaměstnání, kde provozujeme high avaibility řešení pro bankovní a státní instituce, bez žádných zásadních problémů.

Zajímavým námětem mi přijde implementace bonding driver rozšířeného o routovací protokol přímo na serveru.

Reference

Protocol Trickles, *http://www.cs.cornell.edu/~ashieh/trickles/trickles-paper/trickles-nsdi.pdf*

Linux Channel Bonding, *http://sourceforge.net/projects/bonding/*

Cisco Routing Protocols, *http://www.cisco.com/public/technotes/tech_protocol.shtml*

AIX Documentation - EtherChannel and IEEE 802.3ad Link Aggregation,

http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.commadmn/doc/commadmndita/etherchannel_intro.htm

Solaris 10 – IPMP, *http://docs.sun.com/app/docs/doc/816-4554/6maoq027j?a=view*

OpenSolaris – IPMP Rearchitecture, *http://opensolaris.org/os/project/clearview/ipmp/*

Man page ng_one2many(4), *http://www.freebsd.org/cgi/man.cgi?query=ng_one2many&apropos=0&sektion=0&manpath=FreeBSD+6.1-RELEASE&format=html*